

**Clasificación con sistema Bethesda de células escamosas en  
muestras tipo PAP, mediante técnicas de procesamiento de  
imágenes**



**Luis David Medina Sandoval**

**Yamid Eduardo Pineda Sánchez**

**UNIVERSIDAD DEL CAUCA**

**FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

**DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL**

**INGENIERIA EN AUTOMÁTICA INDUSTRIAL**

**POPAYÁN – CAUCA**

**2018**

# **Clasificación con sistema Bethesda de células escamosas en muestras tipo PAP, mediante técnicas de procesamiento de imágenes**



**Monografía presentada como requisito parcial para optar por el título de ingenieros en Automática Industrial**

**Luis David Medina Sandoval**

**Yamid Eduardo Pineda Sánchez**

**Directora: Mg. Elena Muñoz España**

**UNIVERSIDAD DEL CAUCA**

**FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES**

**DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL**

**INGENIERIA EN AUTOMÁTICA INDUSTRIAL**

**POPAYÁN – CAUCA**

**2018**

## **AGRADECIMIENTOS**

En primera instancia agradecemos a los formadores de nuestras vidas, padres de gran sacrificio y esfuerzo, muchos de nuestros logros se los debemos a ustedes entre los que se incluye este. Sus voces de aliento y su infinito amor son nuestros mayores pilares para enfrentar la vida. Asimismo, a nuestros hermanos por estar siempre en todo momento.

Agradecemos también a nuestra asesora de tesis la Mg. Elena España por habernos brindado la oportunidad de recurrir a su conocimiento científico para guiarnos con virtud y paciencia en la realización de nuestro trabajo.

Nuestro agradecimiento también va dirigido para nuestro compañero Jeison Casallas por haber sacrificado parte de su experiencia y tiempo en la construcción del conjunto de imágenes, de igual forma para los laboratorios: Compañía Patólogos del Cauca y Alba Luz por brindarnos las muestras citológicas. También al Dr. Andrés Torres, director de la Unidad de Microscopia de la Universidad del Cauca.

Y finalmente queremos agradecer a nuestros compañeros por habernos entregado su amistad y apoyo moral para seguir adelante con nuestra carrera profesional.

# TABLA DE CONTENIDO

<b>Capítulo 1 Introducción.....</b>	<b>1</b>
1.1. Motivación.....	1
1.2. Estado del Arte .....	2
1.3. Objetivos.....	4
1.4. Resumen de la Propuesta.....	4
1.5. Estructura del Documento .....	5
<b>Capítulo 2 Marco Teórico. ....</b>	<b>6</b>
2.1. Imágenes Histopatológicas.....	6
2.2. Aprendizaje Automático.....	10
2.2.1. Aprendizaje Supervisado .....	11
2.2.2. Aprendizaje No Supervisado .....	12
2.3. Clasificación con Bag of Features.....	12
2.3.1. Detección y Descripción de Características.....	13
2.3.2. Construcción del Codebook.....	29
2.3.3. Representación Bag of Features (BoF).....	31
2.3.4 Clasificación .....	32
<b>Capítulo 3 Clasificación de Imágenes Citológicas.....</b>	<b>41</b>
3.1. Bethesda 2001 .....	41
3.2. Banco de Imágenes Citológicas.....	46
3.3. Esquema Propuesto .....	48
3.3.1. Extracción de Características.....	50
3.3.2. Construcción del Codebook.....	51
3.3.3. Representación de Imágenes.....	53
3.3.4. Clasificación de Imágenes Citológicas .....	54
<b>Capítulo 4 Métricas y Resultados. ....</b>	<b>58</b>
4.1. Métricas de Clasificación .....	58
4.1.1. Matriz de confusión .....	58
4.1.2. Precisión, Recall y F1 .....	59
4.1.3. Área Bajo la Curva ROC .....	60

4.2. Experimentos y Resultados .....	62
4.2.1. Experimento Uno: Dos Clases .....	62
4.2.2. Experimento Dos: Seis Clases Desbalanceadas.....	68
4.2.3. Experimento Tres: Seis Clases Balanceadas .....	75
<b>Capítulo 5 Conclusiones y Trabajos Futuros. ....</b>	<b>81</b>
5.1. Conclusiones.....	81
5.2. Trabajos Futuros.....	82
<b>ANEXOS. ....</b>	<b>83</b>
<b>Capítulo A Manual de Usuario. ....</b>	<b>84</b>
A.1. Instalación y configuración de librerías.....	84
A.2. Manual de Uso .....	96
<b>BIBLIOGRAFÍA.....</b>	<b>105</b>

## ÍNDICE DE TABLAS

Tabla 1. Distribución de imágenes dos clases.....	47
Tabla 2. Distribución de imágenes seis clases desbalanceadas.....	47
Tabla 3. Distribución de imágenes seis clases balanceadas.....	47
Tabla 4. Combinaciones Extractor/Descriptor.....	50
Tabla 5. Configuración de pasos para la búsqueda automático de hiper-parametros. ....	55
Tabla 6. Mejores resultados de la etapa de entrenamiento.....	56
Tabla 7. Tiempo promedio de entrenamiento para cada combinación detector/descriptor. ....	64
Tabla 8. Métricas de rendimiento, precisión, recall y F1 de clasificación binaria.....	65
Tabla 9. Promedio de las métricas para cada combinación. ....	65
Tabla 10. Métricas de rendimiento, precisión, recall y F1 de clasificación multiclase. ....	70
Tabla 11. Promedio de las métricas de clasificación en cada combinación.....	71
Tabla 12. Métricas de rendimiento, precisión, recall y F1 de clasificación multiclase. ....	76
Tabla 13. Promedio de las métricas de clasificación en cada combinación.....	76

## ÍNDICE DE FIGURAS

Figura 1. Anatomía del cuello uterino (fuente: adaptada de [27] ).	7
Figura 2. Estructura del epitelio escamoso en zona exocérvix	7
Figura 3. Células parabasales y basales (fuente: autores).	8
Figura 4. Células intermedias con superposición (fuente: autores).	9
Figura 5. Células intermedias con superposición (fuente: autores).	9
Figura 6. Imagen PAP con tinción de papanicolaou: <b>a.</b> aumento de X40, <b>b.</b> aumento de X100 (fuente: autores).	10
Figura 7. Estructura de aprendizaje supervisado (fuente: adaptado de [30]).	11
Figura 8. Etapas principales de la clasificación de imágenes con el método BoF (Fuente: autores).	13
Figura 9. Filtrado con función gaussiana y desviación estándar ascendente (fuente: autores).	15
Figura 10. Pirámide de diferencias Gaussianas (fuente: adaptada de [11]).	16
Figura 11. Vecindario para evaluación de extremos locales (fuente: adaptada de [11]).	17
Figura 12. Ponderación de las orientaciones de un blob de 16x16 (fuente: autores).	20
Figura 13. Histograma de orientación con picos máximos (fuente: autores).	21
Figura 14. Interpolación parabólica del pico máximo (fuente: autores).	21
Figura 15. Detección de keypoints con el algoritmo SIFT (fuente: autores).	22
Figura 16. Cajas de filtros. Las cajas superiores representan el laplaciano de forma discretizado, mientras las inferiores son una aproximación a estas. Las zonas de color gris equivalen a cero (fuente: adaptada de [42]).	24
Figura 17. Representación del Space-Scale: SIFT (izquierda), SURF (derecha) (fuente: adaptada de [42]).	25
Figura 18. Crecimiento de tamaño de los filtros para las tres primeras octavas (fuente: [42]).	26
Figura 19. Respuestas de filtros Haar-Wavelet (fuente: adaptada de [42]).	27
Figura 20. Asignación de orientación (fuente: adaptada de [42]).	27
Figura 21. Cuadrícula 4x4 sub-regiones (izquierda). Cada sub-divisiones 2x2 se calcula dx, dy, dx y dy (fuente: [42]).	28
Figura 22. Interacción de las fases del algoritmo K-means (fuente: autores).	31
Figura 23. Separación de dos clases en caso linealmente separable (fuente: autores).	35
Figura 24. Transformación de espacio original a espacio de características a través de la función kernel (fuente: adaptada de [66]).	37
Figura 25. Esquema de búsqueda grilla y validación cruzada (fuente: adaptada de [67]).	39
Figura 26. Comparación de multiclasas con: <b>a.</b> OVA y <b>b.</b> OVO (fuente: adaptada de [69] ).	40

Figura 27. Representación morfológica de la clase ASC-US (fuente: autores).....	42
Figura 28. Representación morfológica de la clase L-SIL (fuente: autores). ....	43
Figura 29. Representación morfológica de la clase ASC-H (fuente: autores). ....	44
Figura 30. Representación morfológica de la clase H-SIL (fuente: autores). ....	45
Figura 31. Representación morfológica de la clase Carcinoma (fuente: autores).....	45
Figura 32. Estructura del banco de imágenes (fuente: autores). ....	46
Figura 33. Esquema propuesto de Bag-of-Features (fuente: autores). ....	49
Figura 34. Resultado de la etapa de extracción (fuente: autores). ....	50
Figura 35. Agrupamiento de los descriptores por medio de K-means++ (fuente: autores). ....	52
Figura 36. Ejemplo con 10 clústeres (fuente: autores). ....	54
<i>Figura 37. Estructura de la matriz de confusión (fuente: autores). ....</i>	<i>59</i>
Figura 38. Ejemplo de curva ROC (fuente: adaptada de [74] ). ....	61
Figura 39. Comparación del rendimiento F1 para las combinaciones detector-descriptor (fuente: autores). ....	63
Figura 40. Imágenes utilizadas en evaluación. Las imágenes superiores son anormales y las inferiores de diagnóstico normal (fuente: autores).....	64
Figura 41. <b>a.</b> Comportamiento de la métrica precisión, <b>b.</b> comportamiento de la métrica recall, en un tipo detector/descriptor, SURF/OpponentSURF, variando el valor de K (fuente: autores).....	65
Figura 42. Comportamiento de la métrica F1 en un tipo detector/descriptor, SURF/OpponentSURF, variando el valor de K (fuente: autores).....	66
Figura 43. Matriz de confusión para una combinación SURF/OpponentSURF con un tamaño de codebook igual a 900 (fuente: autores). ....	66
Figura 44. Comportamiento de la métrica ROC AUC de SURF/OpponentSURF, para cada K (fuente: autores).....	67
Figura 45. Curva ROC AUC de SURF/OpponentSURF con K igual a 900 (fuente: autores). ....	67
Figura 46. Comportamiento de las diferentes combinaciones con la métrica F1 (fuente: autores).....	68
Figura 47. Imágenes con células anormales. <b>a.</b> ASCH, <b>b.</b> ASCUS, <b>c.</b> HSIL, <b>d.</b> LSIL, <b>e.</b> Carcinoma (fuente: autores). ....	69
Figura 48. <b>a.</b> Carcinoma con mayor densidad de células superpobladas <b>b.</b> Carcinoma con células dispersas (fuente: autores).....	70
Figura 49. Comportamiento de la métrica F1 en las clases, variando el valor de K para la combinación SIFT/SIFT (fuente: autores). ....	71
Figura 50. Comportamiento de la métrica ROC AUC en SIFT/SIFT, para cada K (fuente: autores).....	72
Figura 51. Curva ROC AUC con un K igual a 500 y una combinación SIFT/SIFT (fuente: autores).....	73
Figura 52. Matriz de confusión de seis clases desbalanceadas (fuente: autores). ....	74
Figura 53. Comportamiento de las combinaciones con la métrica F1 (fuente: autores).....	75

Figura 54. **a.** Comportamiento de la métrica precisión **b.** comportamiento de la métrica recall en un tipo detector/descriptor, SIFT/OpponentSIFT, variando el valor de K (fuente: autores). ..... 77

Figura 55. Comportamiento de la métrica F1 en un tipo detector/descriptor, SIFT/OpponentSIFT, variando el valor de K (fuente: autores). ..... 77

Figura 56. Comportamiento de la métrica AUC en, SIFT/ OpponentSIFT, para cada K (fuente: autores). ..... 78

Figura 57. Matriz de confusión de seis clases balanceadas con una combinación SIFT/OpponentSIFT y un tamaño de codebook a 400 (fuente: autores). ..... 79

Figura 58. Curva ROC AUC con un K=400 y una combinación SIFT/OpponentSIFT (fuente: autores). ..... 80

# GLOSARIO

## E

endocérvix

Parte interna del cuello uterino que forma el canal que conecta la vagina con el útero. La endocérvix está revestida con células que elaboran moco. Durante un examen pélvico se raspan las células de la endocérvix. Las células se observan bajo un microscopio para determinar si hay infección, inflamación o cáncer, o cambios que se pueden volver cancerosos. 18, 60

exocérvix

Es la parte que se ve más fácilmente del cuello uterino a través de la vagina en una colposcopia. Está rodeado por los fondos de saco vaginales. Está recubierto por un epitelio escamoso estratificado rosado, de múltiples capas celulares. Las capas celulares intermedia y superficial del epitelio escamoso contienen glucógeno. 7, 18, 19, 60

## H

hipercromasia

Es un término descriptivo referido a un estado de hipercromacia del núcleo, donde hay una abundancia de ADN que tiñe de oscuro cuando se da tinte para vistas de histología. 56, 57

hipercromático

Aplícase al núcleo que contiene una cantidad anormalmente elevada de pigmentación. 20

hipertrofia

Crecimiento excesivo y anormal de un órgano o de una parte de él debido a un aumento del tamaño de sus células. 58

## M

metaplásicas

Transformación citológica de un epitelio maduro en otro que puede tener un parentesco próximo o remoto. 57

## **N**

neoplasia

Formación anormal en alguna parte del cuerpo de un tejido nuevo de carácter tumoral, benigno o maligno. 55

## **P**

picnóticos

Transformación del núcleo de la célula consistente en una condensación de la cromatina. El núcleo se vuelve homogéneo y se colorea uniformemente. Este fenómeno sería debido a la muerte del núcleo. 20

## **Q**

queratinizante

proceso de formación de queratosis, cambio estructural benigno del tejido escamoso. 19



# Capítulo 1

## Introducción.

---

En este capítulo se presentan los antecedentes de la clasificación de imágenes con un sistema CAD (*Computer Aided Diagnosis*) bajo la metodología de *Bag-of-Features*. En la sección **1.1** se describe la importancia BoF para la clasificación de imágenes, teniendo en cuenta el sistema estándar de diagnóstico en malignidad celular, *Bethesda 2001*. En la sección **1.2** se presentan los diferentes trabajos realizados en el procesamiento de imágenes en problemas de clasificación. En las secciones **1.3** y **1.4** se exponen los objetivos específicos y el resumen del trabajo en cuestión, respectivamente. Finalmente, en la sección **1.5** se establece la estructura del documento.

### 1.1. Motivación

Lograr identificar el grado de malignidad de las células escamosas del epitelio estratificado no queratinizado del cuello uterino en una clasificación estándar, es un reto complejo para el personal especializado debido al criterio subjetivo, ya que las características de las clases precancerosas a simple vista son muy similares. Esta tarea se ha agilizado gracias a la implementación de sistemas automatizados por computador, que logra interpretar y comparar características afines, permitiendo el reconocimiento y clasificación de imágenes, de esta forma dar apoyo cuantitativo a los expertos. La interacción del conjunto de algoritmos adaptados en las fases (preprocesamiento, segmentación, extracción de características, clasificación, e interpretación de imagen), es un punto por investigar, siendo la precisión de la clasificación y el tiempo de procesamiento, las variables de mayor afectación.

Así, pues, la adaptación de algoritmos que permitan disminuir el número fases o la optimización de estos ha llevado a tener métodos alternativos, como *Bag-of-Features* (BoF) o “bolsa de características”. Dentro de las diversas soluciones adaptadas a las etapas de BoF se resaltan los algoritmos denominados operadores de puntos de interés, encargados de extraer y describir la información visual de las imágenes en pequeñas áreas. También, hay técnicas robustas para encontrar automáticamente patrones de representación visual, además de los algoritmos de inteligencia de artificial en apoyo a la clasificación multi-clase.

Actualmente los laboratorios especializados entregan resultados bajo sistemas estandarizados de malignidad celular, siendo el sistema Bethesda versión 2001 el actual para la categorización de las células escamosas de cuello uterino, previamente recolectadas con la prueba de Papanicolau o citología. Bethesda, presenta seis grados de malignidad, resultando su clasificación un desafío para los sistemas automáticos por computador, debido al alto grado de similitud que presenta las características entre dos clases contiguas.

## 1.2. Estado del Arte

El diagnóstico precanceroso de imágenes biomédicas es muy subjetivo y requiere una discusión detallada entre expertos, por esta razón, los sistemas CAD (*Computer Aided Diagnosis*), son utilizados como asistentes de toma de decisiones médicas de segunda opinión.

La mayoría de los sistemas CAD se basan en algoritmos de segmentación que tienen como objetivo principal la identificación del citoplasma y el núcleo, no obstante, estudios recientes [1] demuestran que la detección de la frontera citoplasmática es el principal problema en las imágenes de tejidos escamosos debido a la superposición y el bajo contraste. En la literatura consultada se aborda el anterior problema con múltiples métodos y técnicas, como: la metodología *Watershed* [2], operaciones morfológicas y umbrales automáticos, en conjunto con algoritmos de segmentación jerárquica [3], métodos *star-shape priori* [4], de igual forma han utilizado algoritmos de segmentación jerárquica no paramétrica que manipulan información espectral y de forma [4], sin embargo, los resultados aún presentan condiciones a mejorar, como la delimitación del citoplasma en zonas densas.

Por otra parte, los trabajos basados en extracción de características centran sus objetivos en obtener representaciones matemáticas de la imagen, en un conjunto de vectores con mejor capacidad de discriminación, basadas en los rasgos de textura y morfológicas como: el área, la redondez, perímetro, diámetro corto, elongación, brillo, etc. En [5] presentan la extracción de 160 características, compuestas por tres aspectos: la morfología del núcleo, la patología de la cromatina y la intensidad de la región. Los resultados fueron satisfactorios ya que la textura de la cromatina presenta cambios precancerosos esenciales, permitiendo detectar células anormales. En la extracción de características de texturas, uno de los métodos más utilizados es la matriz de co-ocurrencia de nivel de gris (*Grey level Co-Occurrence Matrix - GLCM*). Dichas características, son mediciones estadísticas como la autocorrelación, la suma de los cuadrados, la variancia, correlación, entre otros, tal y como se muestra en los trabajos [6][7]. La extracción de características morfológicas o geométricas, involucra la fase de segmentación, siendo esta una desventaja con respecto a las técnicas de detección de puntos de interés. Las técnicas basadas en detección de puntos de interés (*keypoints*) no requieren realizar un proceso de segmentación, las imágenes se describen con vectores de

características invariantes a factores de rotación, escalado y parcialmente a cambios de iluminación, siendo la transformación de característica invariante de escala (SIFT) el algoritmo más representativo. En [8] utilizan este tipo de detector y descriptor para realizar la anotación automática de imágenes médicas, es decir la tarea de asociar conceptos semánticos que están presentes en una imagen. Finalmente, afirman que es posible adaptar SIFT para resolver problemas de anotación a partir de un esquema sencillo. En el trabajo de Caicedo et al [9] desarrollaron una evaluación de dos representaciones obtenidas del enfoque de BoF para clasificar imágenes histopatológicas, donde se detectaron y describieron las características con dos estrategias. La primera es *dense random sampling* [10], el cual selecciona puntos en el plano de la imagen al azar y luego, define un blob de píxeles alrededor de la coordenada del punto. La segunda estrategia se basa SIFT [11] que calcula descriptores invariantes de rotación, utilizando orientaciones predefinidas sobre un conjunto de bloques. El algoritmo SIFT, es ampliamente utilizado bajo el enfoque *Bag-of-Features* (BoF), este no requiere conocimiento experto y en su lugar aprende un libro de códigos visuales o *codebook*, con características morfológicas [12][13].

La metodología BoF es un esquema alternativo para la clasificación y recuperación de información de imágenes conformado por cuatro fases[14][8][15]. En el procesamiento en imágenes biomédicas, se han aplicado con éxito en el cribado de cáncer, por ejemplo, Raza et al [16] desarrollaron un sistema de clasificación en subcategorías del carcinoma de células renales (RCC). Las principales contribuciones de este enfoque son la ausencia de segmentación de color y no requieren imágenes de plantillas elegidas por expertos. En [17] se presenta una implementación de sistema de diagnóstico asistido por computador para el cribado precoz de melanomas, bajo un enfoque BoF. Realizaron una comparación de descriptores de textura y color de forma independiente, donde las dos mejores técnicas por cada categoría fueron utilizadas para concatenar sus respectivos vectores de características. Los resultados demuestran una mejor clasificación de las imágenes de piel. Sin embargo, el esquema BoF tiene pocos estudios de clasificación en imágenes citológicas.

La creación del libro de códigos visuales se obtiene mediante una metodología de agrupamiento donde la mayoría de investigaciones están centradas en el algoritmo *K-means* y sus derivados como *K-means++* [18], *mean-shift* [19] y jerárquica *K-means* [20]. Tomassi et. al [21] demostró que el tamaño del diccionario visual no es un aspecto significativo en la tarea de clasificación de imágenes de radiología.

Finalmente, la clasificación divide las imágenes histopatológicas en las categorías de diagnóstico precancerosas. La clasificación de imágenes con BoF se basa en el aprendizaje de las representaciones del banco de imágenes de entrenamiento, obteniendo un modelo o condiciones de separación.

En [9] se presenta una evaluación de diferentes representaciones BoF para clasificar imágenes patológicas a través del kernel: *intersección del histograma*, además, de una función de base radial compuesta (RBF). En el trabajo de Kasemsit et al [22] exploran varios

clasificadores basados en máquinas de vectores de soporte (SVM) como: máquinas de vectores de soporte doble (TWSVM) y máquinas de vectores de soporte de doble hiper-esfera (THSVM), donde se muestra el desempeño en la clasificación de células de cáncer cervical. Asimismo se encuentra variada literatura de SVM con buenos resultados en diferentes métricas de rendimiento [14][16][23], no obstante se han utilizado otros clasificadores para diagnosticar cáncer cervical, en detalle son redes neuronales artificiales (RNA) , vecino más cercano (NN) [24], y árboles de decisión [25].

A pesar de que la metodología BoF es flexible y presenta muchas ventajas, aún sigue siendo objeto de estudio en imágenes histopatológicas, de igual forma se avanza en el mejoramiento en cada una de sus etapas con diferentes algoritmos. Siendo la extracción de características y la técnica de clasificación, las etapas de mayor interés y contribución al estado técnico actual.

### 1.3. Objetivos

Proponer un sistema para clasificar las anomalías de las células escamosas en zona exocérvical bajo el sistema *Bethesda 2001*, es el objetivo principal de este proyecto, de forma que agilice el trámite de la entrega de resultados con un correcto cribado, para establecer el seguimiento y tratamiento de la paciente.

Los objetivos específicos que se pretenden cumplir con este trabajo son:

1. Construir un sistema basado en técnicas de procesamiento de imágenes digitales para determinar anomalías de células escamosas.
2. Implementar un sistema de clasificación utilizando la técnica *Bag of Features* con muestras del laboratorio de Citología de la Universidad del Cauca.
3. Evaluar de forma cuantitativa el error de detección del sistema propuesto, respecto a los diagnósticos clínicos de una muestra de pacientes del laboratorio de Citología de la Universidad del Cauca.

### 1.4. Resumen de la Propuesta

El trabajo propone un sistema de clasificación de imágenes PAP (Papinocolau) para determinar el grado de malignidad según el sistema estandarizado Bethesda, versión 2001.

Su centro de atención es la implementación y unificación de los algoritmos más representativos de cada fase de la metodología de *Bag of Features*. En cada una de las fases se implementa algoritmos ligeros y robustos para establecer un sistema de *Diagnostico*

*Asistido por Computador (CAD)* con un modelo o condiciones de clasificación precisa y rápida.

El desempeño de la clasificación se evalúa de forma cualitativa y cuantitativa apoyado de métricas especializadas permitiendo hacer una comparación de los algoritmos, además de obtener resultados que puedan ser analizados con el criterio del personal experto y así poder determinar la utilidad del sistema en el cribado de malignidad celular.

El desarrollo del sistema ha sido implementado en el lenguaje multiparadigma Python con apoyo de las siguientes librerías: OpenCV, numpy, sklearn, scipy, matplotlib, y la herramienta PyQt4 para el desarrollo de la interfaz gráfica.

## 1.5. Estructura del Documento

El trabajo consta de cinco capítulos (incluido el presente), establecidos en una posición coherente para dar claridad a los temas propuestos. Estos van desde el marco teórico hasta la implementación, para finalizar con la validación del desempeño del algoritmo, conclusiones y trabajos futuros. En el **Capítulo 2**, se detallan las definiciones de los diferentes temas que abarcan este trabajo, partiendo de una breve descripción de las imágenes histopatológicas hasta las fases de la metodología *Bag of Features*. De esta manera se abordan los detalles técnicos en la clasificación de imágenes histopatológicas de muestras PAP. El **Capítulo 3** da a conocer en detalle la configuración de los diferentes algoritmos implementados en cada una de las fases del sistema de clasificación. El **Capítulo 4**, presenta la definición y validación experimental de la propuesta por medio de cinco métricas, para tres conjuntos diferentes de imágenes. Finalmente, el **Capítulo 5**, presenta las conclusiones de la aplicación del sistema de clasificación implementado en un enfoque de imágenes citológicas, de igual forma se propone mejoras para trabajos futuros.

# Capítulo 2

## Marco Teórico.

---

En el presente capítulo se expone la teoría, limitada desde la importancia de las imágenes histopatológicas hasta el marco de la metodología de *Bag of Features* para clasificarlas en seis clases, por esta razón se contemplan las siguientes secciones. En la sección **2.1** se da a conocer en términos generales la anatomía del cuello uterino y las dos condiciones para facilitar el análisis de imágenes PAP. En la sección **2.2** se expone los tipos de aprendizaje automático (supervisado y no supervisado) para los algoritmos de clasificación. En la última sección **2.3** se contextualiza el método BoF con sus respectivos algoritmos de extracción/descripción de información visual, la creación del *codebook*, la representación y finalmente la clasificación.

### 2.1. Imágenes Histopatológicas

La histopatología estudia las diferentes células y tejidos de un cuerpo, para determinar alguna anomalía en la estructura biológica. Esta rama de la patología es utilizada para diagnosticar el grado de malignidad de las muestras citológicas, es decir, detectar las lesiones precancerosas del tejido escamoso en el cuello uterino. Dichas muestras son recolectadas en dos zonas de la anatomía del cuello, *exocérvix* y *endocérvix* (ver Figura 1), por medio de la prueba de citología o Papanicolau, para luego ser extendidas sobre un portaobjetos [26].

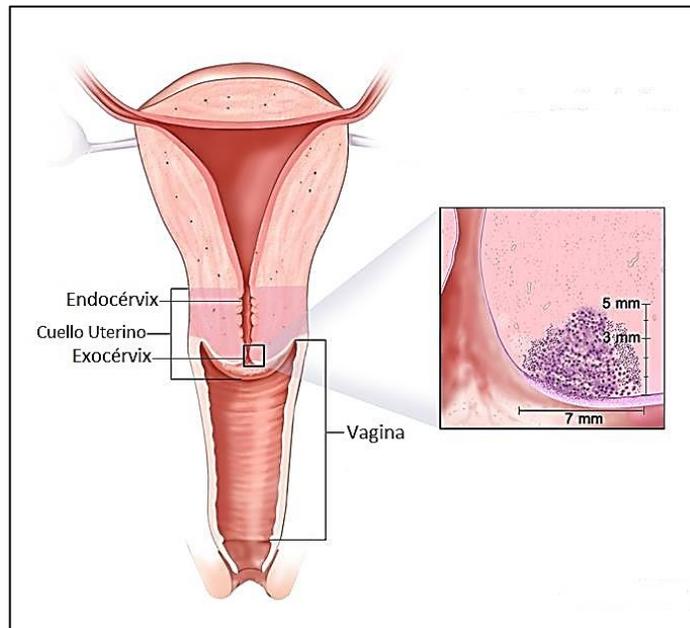


Figura 1. Anatomía del cuello uterino (fuente: adaptada de [27] ).

El estudio detallado por el personal especializado requiere de una buena tinción y la magnificación, por lo tanto, estas dos condiciones son importantes para obtener imágenes de buena calidad. La tinción permite resaltar el contraste de la estructura biológica de las cuatro clases de células (basales, parabasales, intermedias y superficiales), divididas en dos partes (inmadura y madura) del epitelio escamoso no queratinizante (ver Figura 2).

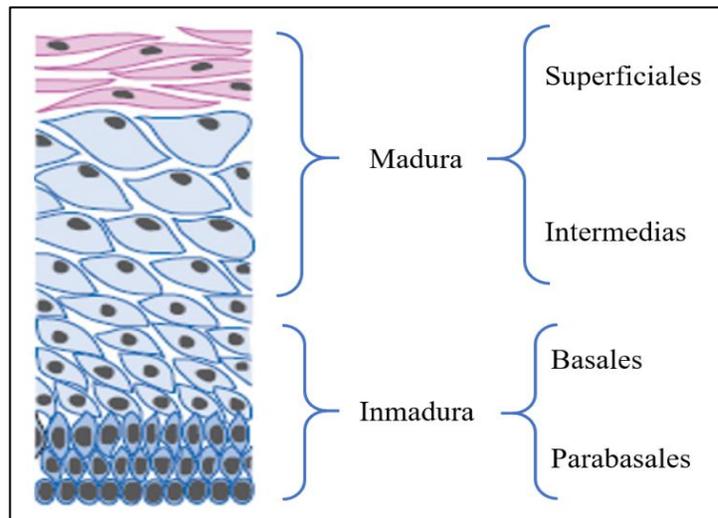
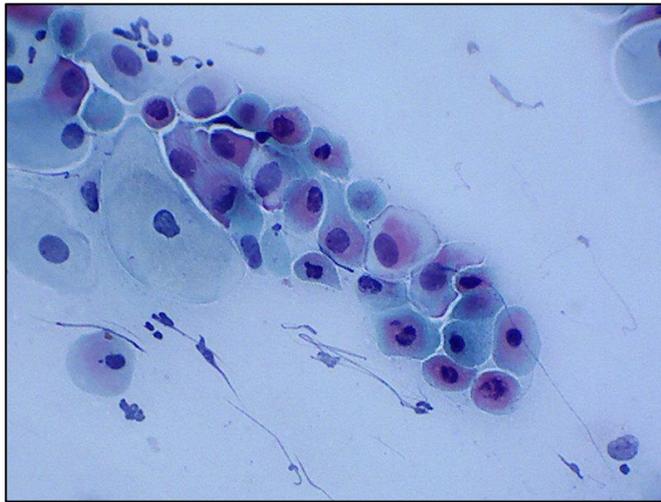


Figura 2. Estructura del epitelio escamoso en zona exocervix (fuente: autores).

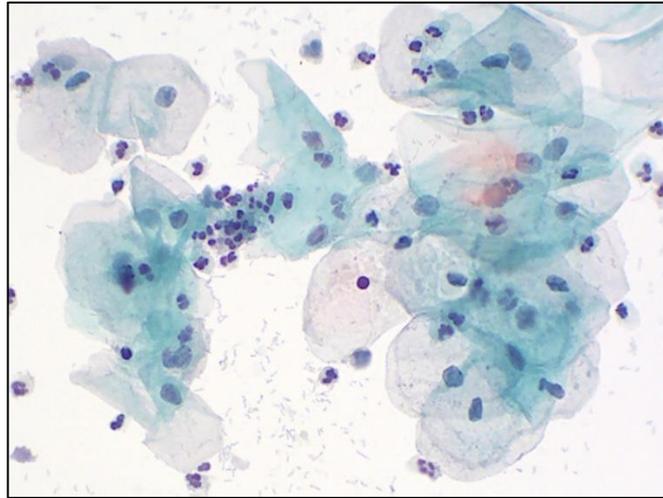
**Células Parabasales:** Estas aparecen en los estados atróficos de la infancia y menopausia; en cuanto a los aspectos morfológicos son células pequeñas de (15 – 30 micras), elípticas, con mayor frecuencia de citoplasma con tinción azul y sus bordes bien definidos. Los núcleos son redondos y ovales, miden entre (8 – 12 micras) y en ocasiones por consecuencia de una maduración rápida pueden presentar núcleos pequeños y picnóticos.

**Células Basales:** Son células de menor tamaño a las anteriores, entre (14 – 20 micras), con un núcleo que es central, redondo e hiper cromático; el citoplasma al igual que las células parabasales es de tinción azul, con borde escaso y definidos (ver Figura 3).



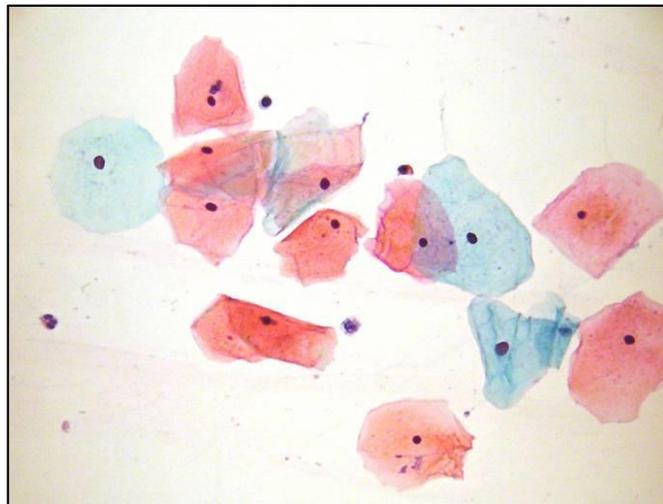
*Figura 3. Células parabasales y basales (fuente: autores).*

**Células Intermedias:** se originan en el estrato medio del epitelio y son las células que más se presentan en la fase post-ovulatoria. Son células grandes entre (30 – 50 micras), de citoplasma transparente, poligonal y de bordes plegados; su tinción toma un tono verde-azulado y los núcleos son redondos u ovales pero mayores que los de las células superficiales miden entre (9 – 11 micras) (ver Figura 4).



*Figura 4. Células intermedias con superposición (fuente: autores).*

**Células Superficiales:** Como su nombre lo indica se originan en el tejido superficial del epitelio, son las más comunes en la etapa preovulatoria y poseen el mayor grado de madurez. Son células grandes miden entre (40 – 60 micras), con forma poligonal, de citoplasma translucido con bordes bien definidos e irregulares y con picnosis independiente de la tinción del citoplasma que suele ser rosada; su núcleo es pequeño entre (5 – 7 micras) de forma redonda y central (ver Figura 5).



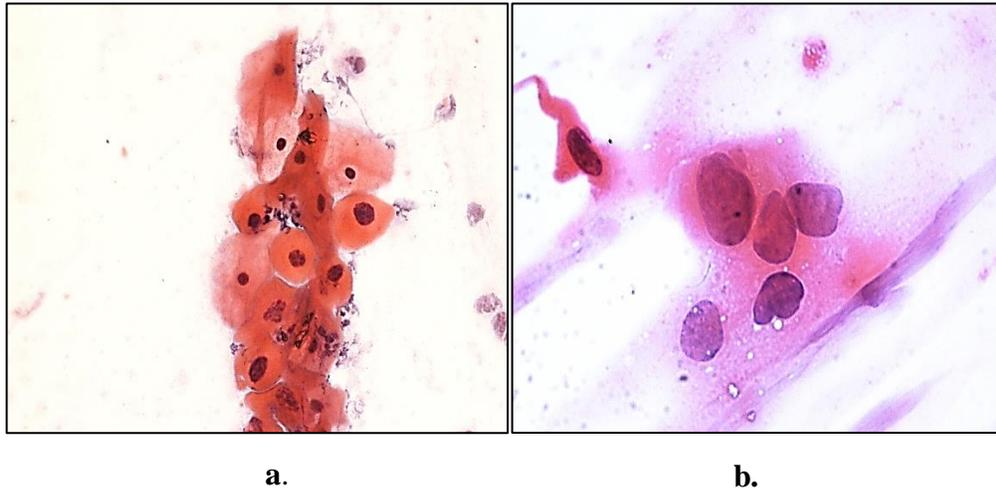
*Figura 5. Células intermedias con superposición (fuente: autores).*

Existen diversos tipos de tinción, el cual es determinado por el material biológico a contrastar. En las muestras de citología se utiliza la tinción *Papanicolaou*, introducida por el médico griego Georgios Papanicolaou en 1923. Esta permite conservar las células con una fijación y color adecuado (ver Figura 6).

## 2.2. Aprendizaje Automático

---

Durante el análisis con el microscopio, es fundamental establecer la magnificación, la cual hace referencia a la proporción de aumento de las estructuras biológicas, determinadas por el conjunto de objetivos o lentes. La mayoría de los microscopios poseen objetivos convencionales de 4X, 10X, 40X, y 100X. El objetivo de 40x garantiza distinguir la composición de las células en un campo considerable, por otra parte, el de 100X abarca una pequeña área, pero con mayor nitidez en las células.



*Figura 6. Imagen PAP con tinción de papanicolaou: a. aumento de X40, b. aumento de X100 (fuente: autores).*

Por lo tanto, estos dos objetivos son ideales para el registro de imágenes histopatológicas [26][28].

## 2.2. Aprendizaje Automático

Es el campo de la inteligencia artificial que busca el aprendizaje de las computadoras en forma automática para determinar o clasificar la dinámica de cierto fenómeno. En 1997 Tom Mitchell lo define como la capacidad que tiene un programa de computadora para mejorar su rendimiento en alguna tarea a través de la experiencia. A continuación, se presenta la definición exacta introducida por Tom Mitchell,

*“Se dice que un programa de computadora aprende de la experiencia  $E$  con respecto a alguna clase de tareas  $T$  y la medida de rendimiento  $P$ , si su desempeño en tareas en  $T$ , medido por  $P$ , mejora con la experiencia  $E$ ”.*

Las tareas se dividen generalmente en dos grupos (*aprendizaje supervisado* y *no supervisado*) que depende de la condición de etiqueta para cada muestra del conjunto de datos de entrada, no obstante, existen diversas metodologías de aprendizaje [29].

### 2.2.1. Aprendizaje Supervisado

El aprendizaje supervisado es aquel donde se tiene un conjunto de datos con su respectiva etiqueta, o salida deseada para calcular un modelo o función que permite dar solución a cada dato de evaluación.

La definición formal está dada por un conjunto de pares, conformado por las variables de entrada,  $x^{(i)}$ , y las variables de salida u objetivo,  $y^{(i)}$ . De esta manera se obtiene el conjunto de entrenamiento para un algoritmo, con  $m$  muestras,  $\{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}$ . En un contexto real, generalmente las variables vienen representadas por vectores, por lo tanto,  $\chi$  y  $\mathcal{Y}$  son los vectores de entrada y salida respectivamente. De esta forma se denota  $h : \chi \rightarrow \mathcal{Y}$ , como la función que permite aproximar el conjunto de entradas al de salidas, que comúnmente se denomina *hipótesis*. En la Figura 7 se ilustra el proceso:

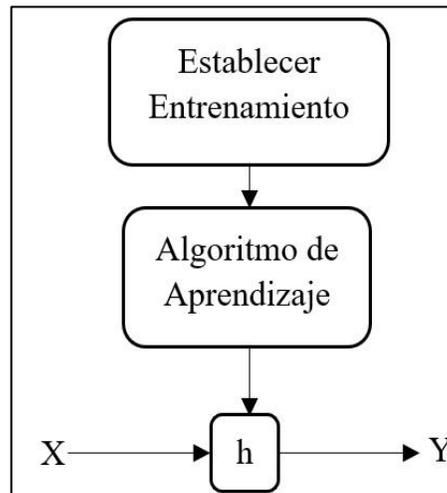


Figura 7. Estructura de aprendizaje supervisado (fuente: adaptado de [30]).

En un enfoque de procesamiento de imágenes las entradas son descritas por múltiples características, es decir se representa como un vector de  $n$  dimensiones,  $x^{(i)} \in \mathbb{R}^n$ . La finalidad de esta metodología de aprendizaje es predecir la variable de salida. Cuando se tiene como objetivo predecir valores continuos el problema es denominado *regresión* o si se presenta la tarea de asignar una clase, es decir predecir a que clase pertenece un conjunto de datos se resolverá un problema de *clasificación*. Entre los métodos más conocidos de

aprendizaje supervisado están las máquinas de soporte vectorial (SVM), redes neuronales artificiales (RNAs) y el clasificador k vecinos más cercanos.

Mustafa et al [31] utilizan SVM para clasificar las lesiones de piel presentes en imágenes cancerosas y no cancerosas, obtienen resultados de exactitud del 80% para ambas clases. Por otro lado, Kieffer et al [32] hacen una comparación del entrenamiento y uso de redes pre-entrenadas en el algoritmo de redes neuronales convolucionales para clasificar imágenes histopatológicas, concluyen que las redes pre-entrenadas presentan un mejor desempeño de clasificación respecto al entrenamiento desde cero.

### 2.2.2 Aprendizaje No Supervisado

Esta metodología de aprendizaje se refiere cuando el conjunto de entradas no tiene una etiqueta asociada, es decir, el algoritmo realiza el análisis exploratorio de datos para encontrar el modelo o patrones ocultos, que generalmente está determinado por el agrupamiento de las muestras en *clústeres*. Un clúster es una colección de objetos similares definido por una medida en un espacio multidimensional. Por lo tanto, el aprendizaje no supervisado se centra en problemas de *clustering*. Este enfoque se utiliza para separar los vectores característicos de una imagen en  $K$  clústeres durante el proceso de entrenamiento con el fin de representar un grupo de imágenes por medio de un histograma, tal como lo realiza el método BoF.

Goswami et al [33] presentan una técnica de red neuronal basada en el aprendizaje no supervisado para la clasificación de imágenes de resonancia magnética del cerebro humano por medio de un procedimiento de tres fases. La primera, comprende el preprocesamiento de las imágenes con la técnica de ecualización de histogramas, detección de bordes, etc. La segunda realiza la extracción de características usando el análisis de componentes independientes (ICA). En la tercera fase, se diagnostica el tumor cerebral mediante el mapa autoorganizado (SOM). Finalmente, aplican un algoritmo de agrupamiento *K-means* para segmentar el cerebro en diferentes tejidos.

En el trabajo de Roy et al [34] presentan una clasificación de imágenes a través del algoritmo de agrupamiento no supervisado, Teoría de Resonancia Adaptativa (ART), el cual es una de las técnicas alternativas de *K-means*. La ventaja de ART es la no indicación previa del número de clústeres. Esta ventaja se exploró para agrupar las características extraídas y descritas por SURF, en un contexto BoF.

## 2.3 Clasificación con Bag of Features

*Bag-of-Features* o “*Bolsa de Características*”, es un método alternativo para la

representación de imágenes, enfocado en solucionar problemas de recuperación y clasificación [35][36][37]. Está basado en el modelo *Bag-of-Words* o “*bolsa de palabras*” que tiene como fin la diferenciación de diversos tipos de documentos, a través de la identificación del vocabulario de palabras más significativas. En el contexto del procesamiento de imágenes, bajo un enfoque BoF estas son representadas por un conjunto de parches visuales. El marco de clasificación de imágenes basado en la metodología de BoF presenta un esquema continuo de cuatro etapas [38][39][40]: Detección y descripción de características, Construcción del *codebook*, Representación BoF, y Clasificación. En la Figura 8 se resume la secuencia de las etapas.

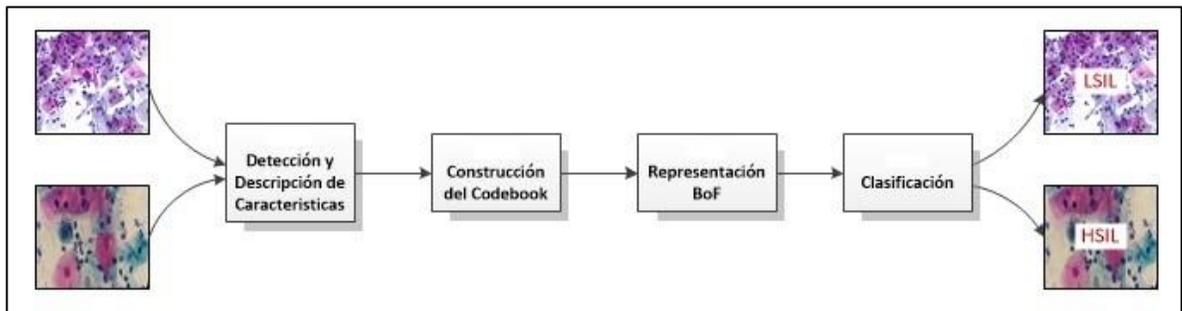


Figura 8. Etapas principales de la clasificación de imágenes con el método BoF (Fuente: autores).

Los parches visuales calculados son representados en un vector de información robusta, es decir discriminativo y fiable. Esto garantiza obtener una representación suficiente para encontrar diferenciación o coincidencias de las imágenes. La representación es definida por la cuantificación de parches del mismo tipo, creando así un histograma “visual”. Esto es ideal para procesar grandes bancos de imágenes en un tiempo reducido.

### 2.3.1 Detección y Descripción de Características

El objetivo es la búsqueda de características o rasgos distintivos por medio de técnicas basadas en características locales y globales. Dichas técnicas, se enfocan en la detección de puntos de interés que contengan información relevante para obtener una descripción discriminante con referencia al vecindario. La información se representa en un vector de características denominado “*Descriptor*”. Para encontrar similitudes y llegar a clasificar las imágenes, los descriptores serán comparados usando una medida de similitud resultando en un factor de correlación de los *Keypoints*, sin importar la rotación, escalado, cambios de iluminación o ruido de la imagen [41]. Esto ha llevado en la última década al desarrollo de técnicas y algoritmos que cumplan dichas condiciones, como, por ejemplo: SIFT [11],

SURF[42], ORB [43], FAST [44], BRIEF [45], FREAK [46], BRISK[47]. Sin embargo, en esta propuesta solo se describen los algoritmos SIFT y SURF debido al buen desempeño de procesamiento de imágenes [9] [48] [24] [49] [50].

### 2.3.1.1 Scale-Invariant Feature Transform (SIFT)

Es una técnica diseñada por David Lowe, donde las características extraídas de las imágenes son invariantes a escala, rotación, y parcialmente a cambios de iluminación. El costo computacional en extraer dichas características se minimiza mediante un enfoque de filtrado en cascada, en el cual las operaciones de mayor procesamiento se llevan a cabo en lugares que pasan una prueba inicial. Un aspecto importante de este enfoque es la coincidencia y comparación de imágenes, por lo tanto, las características SIFT se extraen de un conjunto de imágenes almacenadas en una base de datos. La coincidencia, se lleva a cabo por medio de la comparación individual de las características de una nueva imagen de prueba, con el conjunto de características obtenido. Finalmente, la coincidencia está determinada por la distancia euclidiana de sus vectores. A continuación, se explica las cuatro etapas principales presentadas en [11].

#### 1. Detección de extremos en el Espacio-Escala

El fin principal de la etapa de detección es identificar las ubicaciones y las escalas asignadas repetidamente bajo las diferentes vistas del mismo objeto, esto hace referencia a que las ubicaciones deben ser invariantes al escalado, translación y rotación de una misma imagen. Para obtener una descripción de un mismo objeto ante variaciones de escala se utiliza la función *Scale-Space* [51] que consiste en el filtrado progresivo de la imagen,  $I(x, y)$ , para eliminar detalles, creando una familia de imágenes que son derivadas de la convolución con la función gaussiana,  $G(x, y, \sigma)$  [52][53]. Por lo tanto, *Scale-Space*, representado por  $L(x, y, \sigma)$ , está definida por la ecuación 1.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (1)$$

Donde

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2)$$

Los parámetros  $x, y$  indican la coordenada de cada pixel y  $\sigma$  es la desviación estándar. Este último parámetro controla el grado de suavización de la imagen. La Figura 9 ilustra lo anterior.

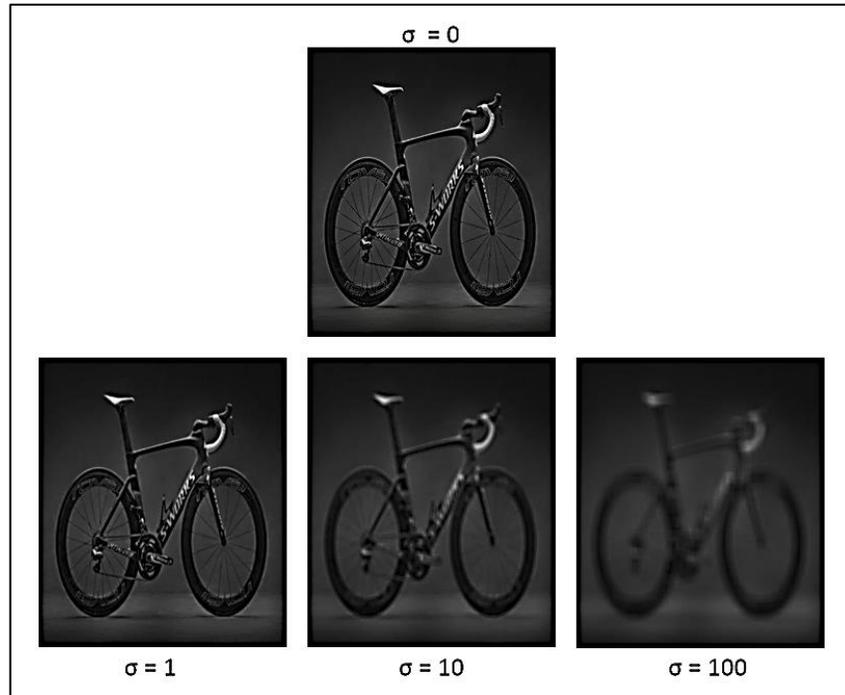


Figura 9. Filtrado con función gaussiana y desviación estándar ascendente (fuente: autores).

Una vez obtenida la familia de imágenes con diferentes índices de desviación estándar, se busca los puntos característicos que cumplan la invariancia al escalado, a través de la función de las diferencias Gaussianas (DoG), denotada por  $D(x, y, \sigma)$ , y dentro de una pirámide (ver Figura 10)

$$\begin{aligned}
 D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\
 &= L(x, y, k\sigma) - L(x, y, \sigma)
 \end{aligned}
 \tag{3}$$

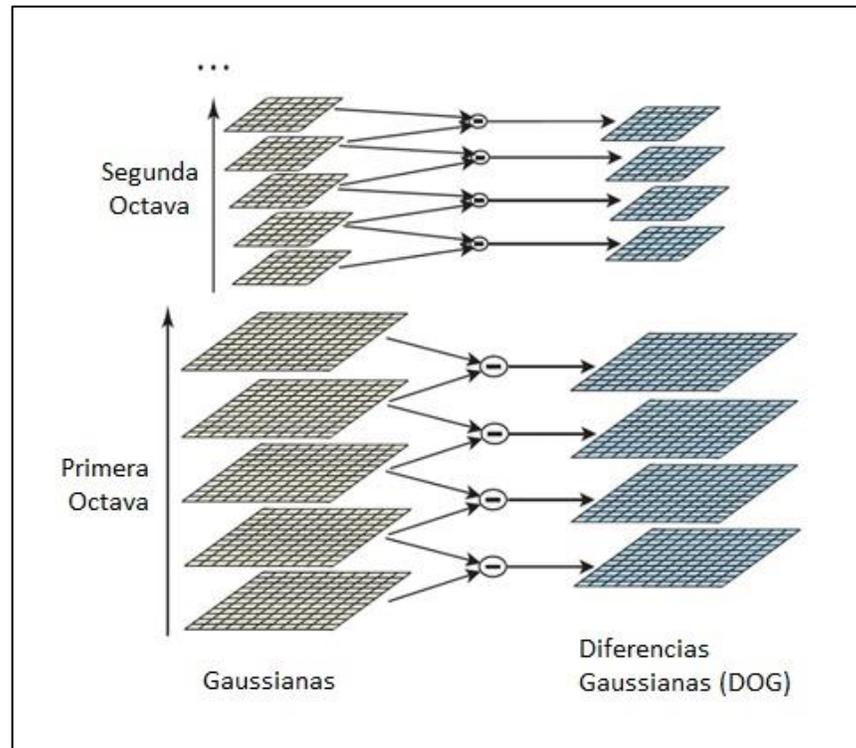


Figura 10. Pirámide de diferencias Gaussianas (fuente: adaptada de [11]).

La columna de la izquierda de la Figura 10 es conocida como octava y corresponde a la familia de imágenes conseguidas por la convolución gaussiana con valores ascendentes de desviación estándar, generando diferentes imágenes con grado de suavización, denominadas escalas [11]. El número de octavas y escalas suelen estar relacionadas con el tamaño de la imagen, sin embargo, Lowe sugiere en su trabajo de 2004, que 4 octavas y 5 escalas son suficientes para la detección de características invariantes. Cada octava está dividida en intervalos múltiples de  $K$ , además la desviación estándar debe cumplir la condición: la cuarta escala debe ser el doble de la primera ( $\sigma_4 = 2\sigma_1$ ) [54]. Las ecuaciones 4 y 5 fundamentan lo anterior.

$$K = 2^{\left(\frac{1}{\# \text{escalas} - 2}\right)} \quad (4)$$

$$\sigma_i = K^{i-1} = 2^{\left(\frac{i-1}{3}\right)} \quad (5)$$

Con una octava completa, se aplica la función DoG a las escalas adyacentes, formando la pila derecha de escalas DoG (ver Figura 10), utilizadas para calcular los extremos locales,

(explicado en la siguiente sección). Por último, se re-muestra la escala que tiene el doble del valor inicial de  $\sigma$ , tomando dos píxeles en cada fila y columna, creando la siguiente octava con la mitad del tamaño de las escalas de la anterior.

### 1.1 Detección de Extremos Locales

La detección de los máximos y mínimos locales de  $D(x, y, \sigma)$  se lleva a cabo por una comparación de cada punto de la muestra, con sus ocho vecinos de la imagen actual y los nueve vecinos de la escala superior e inferior (ver Figura 11). El punto es considerado extremo solo si el valor de intensidad es mayor que el resto de todos sus 25 vecinos o es menor que todos ellos. Este proceso es de bajo costo debido a que la mayoría de los candidatos son eliminados en los primeros chequeos.

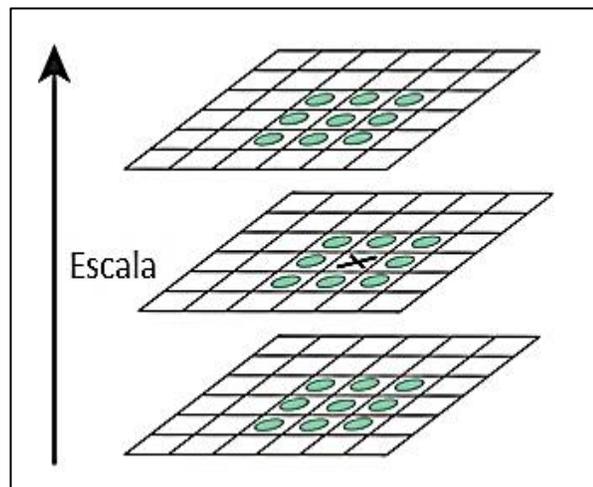


Figura 11. Vecindario para evaluación de extremos locales (fuente: adaptada de [11]).

Por lo tanto, las escalas 2 y 3 de la pila de DoG de cada octava, son las únicas que se utilizan para la búsqueda de máximos o mínimos, ya que solamente estas comparten una escala vecina de nivel inferior y superior (ver Figura 10).

## 2. Localización Precisa del Punto Clave

En esta etapa se realiza un ajuste detallado de los datos cercanos, correspondientes a la comparación del punto clave con sus vecinos, para ello se determina la ubicación, escala y relación de curvaturas principales. Esta información rechaza puntos de bajo contraste, siendo estos sensibles al ruido, y aquellos que se encuentren en una localización a lo largo de un borde.

### 2.1. Eliminación de Puntos de Bajo Contraste

En el método de Brown et al [55] se ajusta una función cuadrática 3D a los puntos de una muestra local con el fin de determinar la localización interpolada del máximo. Esto conlleva a una mejora sustancial en la búsqueda de correspondencias y la estabilidad. El método se centra en la utilización de la expansión de Taylor hasta los términos cuadráticos de la función de *Scale-Space* en  $D(x, y, \sigma)$ , desplazada de forma que el origen esté en el punto de la muestra (Ecuación 6).

$$D(\chi) = D + \frac{\partial D^T}{\partial \chi} \chi + \frac{1}{2} \chi^T \frac{\partial^2 D}{\partial \chi^2} \chi \quad (6)$$

Donde  $D$  y sus derivadas son evaluadas en el punto de muestra y  $\chi = (x, y, \sigma)^T$  es el *offset* desde ese punto. La ubicación del extremo,  $\hat{\chi}$ , se obtiene por medio de la ecuación 7, donde se calculan las derivadas de la función con respecto a  $\chi$  e igualando a cero [11].

$$\hat{\chi} = - \frac{\partial^2 D^{-1}}{\partial \chi^2} \frac{\partial D}{\partial \chi} \quad (7)$$

si  $\hat{\chi} > 0.5$  en cualquier dimensión, significa que el extremo se encuentra más cercano a un punto de muestra diferente por lo tanto la interpolación se realiza en ese punto.

Finalmente, se calcula el valor de extremo sustituyendo la ecuación 7 en 6. Esto permite rechazar puntos inestables con bajo contraste.

$$D(\hat{\chi}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \chi} \hat{\chi} \quad (8)$$

Los experimentos de Lowe concluyeron que si  $|D(\hat{\chi})| > 0.03$ , serán descartados, suponiendo que el valor de los pixeles es de rango  $[0, 1]$ .

### 2.2. Eliminación de Puntos en Bordes

Un extremo ubicado en un borde presentará una gran curvatura a lo largo de este y una pequeña curvatura en dirección perpendicular, conllevando a la inestabilidad. Dicha

curvatura es determinada por medio de la matriz hessiana de 2x2 (ecuacion 9), en la localizacion y escala del punto.

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \quad (9)$$

Los valores propios ( $\alpha$   $\beta$ ) de H son proporcionales a las curvaturas principales de D, esto implica trabajar con la traza y el determinante de la matriz, definidas por las ecuaciones 10 y 11 respectivamente.

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta \quad (10)$$

$$Det(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha \beta \quad (11)$$

El punto será descartado si el determinante es negativo, ya que las curvaturas son de diferente signo, de igual forma se eliminan los puntos, con curvas mayores al umbral de relación entre  $\alpha$  y  $\beta$ . Considerando que la relación entre ambos valores propios es ( $\alpha = r\beta$ ), Lowe propone evaluar la siguiente expresión y conservar solo los puntos que la cumplan

$$\frac{Tr(H)^2}{Det(H)} < \frac{(r + 1)^2}{r}$$

### 3. Asignación de Orientación

A los *keypoints* con invariancia de escalado y rotación determinados en las imágenes Gaussianas,  $L(x, y)$ , se les define una región alrededor, denominada *blob*, con dimensión de 16 x 16 muestras, divida en 4 regiones. A cada celda se determinan la orientación,  $\theta(x, y)$ , y su correspondiente Magnitud del gradiente,  $m(x, y)$ , calculadas mediante las ecuaciones 12 y 13 respectivamente.

$$\theta(x, y) = \tan^{-1} \left( \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \quad (12)$$

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2} \quad (13)$$

## 2.3 Clasificación con Bag of Features

Con cada orientación del *blob*, se forma un histograma, los cuales son ponderados por su magnitud del gradiente  $m(x, y)$ , y por una ventana circular gaussiana con valor de desviación estándar,  $\sigma$ , igual a 1.5 veces la escala del *keypoint*. La Figura 12, ilustra lo anterior.

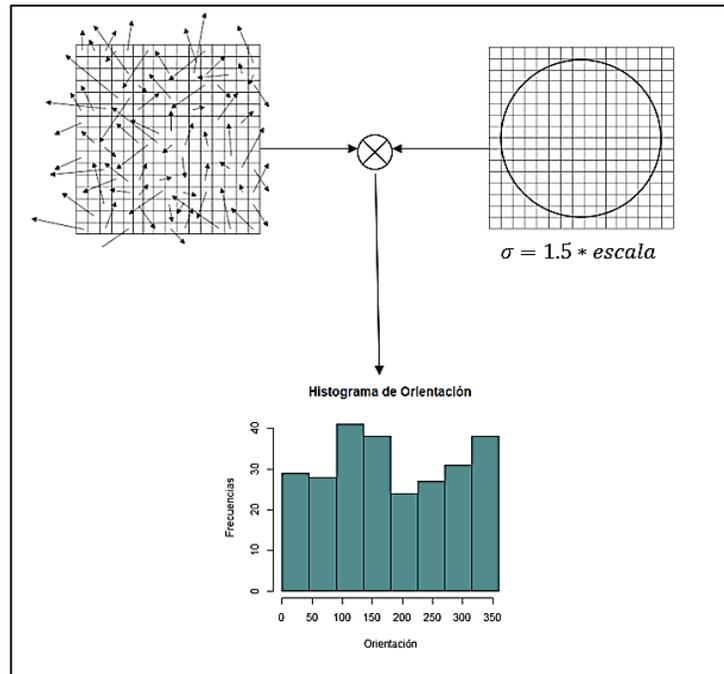


Figura 12. Ponderación de las orientaciones de un blob de 16x16 (fuente: autores).

La ponderación de la magnitud del gradiente y la ventana circular gaussiana asigna mayor peso a las orientaciones con más valor de  $m(x, y)$ , que por tanto son más importantes. Los picos más altos en el histograma corresponden a las direcciones dominantes, determinando la orientación final del punto clave. Por lo tanto, se detecta el pico más alto, y luego cualquier otro pico local que este dentro del 80% del pico más alto, creando un nuevo punto clave con esa orientación, es decir, habrá múltiples *keypoints* creados en la misma ubicación y escala, pero con diferentes orientaciones (ver Figura 13), estos contribuyen a la estabilidad de la coincidencia.

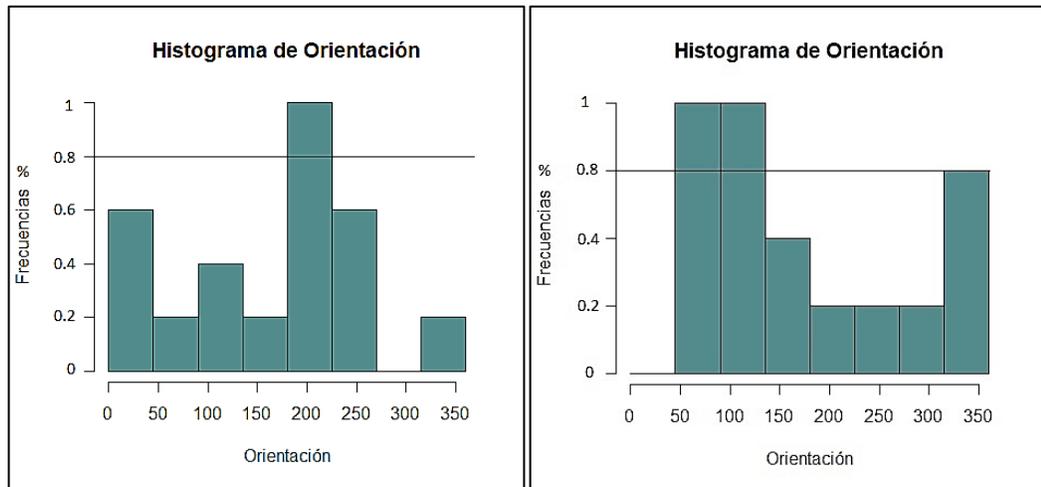


Figura 13. Histograma de orientación con picos máximos (fuente: autores).

Finalmente, para lograr una mayor precisión se lleva a cabo la interpolación de su posición mediante la construcción de una parábola con los tres valores más cercanos a cada pico seleccionado (ver Figura 14).

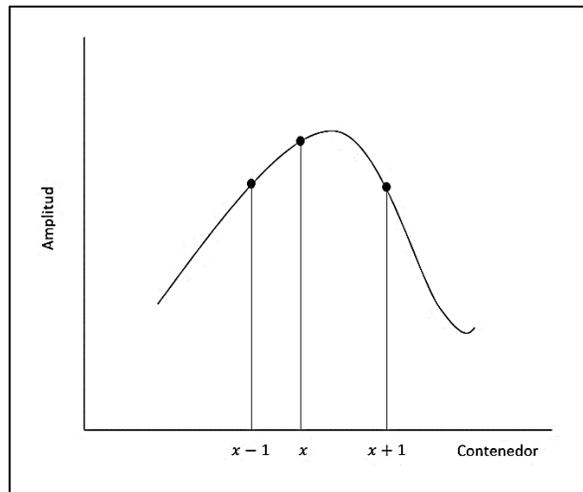


Figura 14. Interpolación parabólica del pico máximo (fuente: autores).

La Figura 15 muestra la respuesta del algoritmo SIFT en la detección de *keypoints*, los cuales son representados por una región circular de diferente radio, esta diferencia es debida al espacio-escala en el cual es detectado.

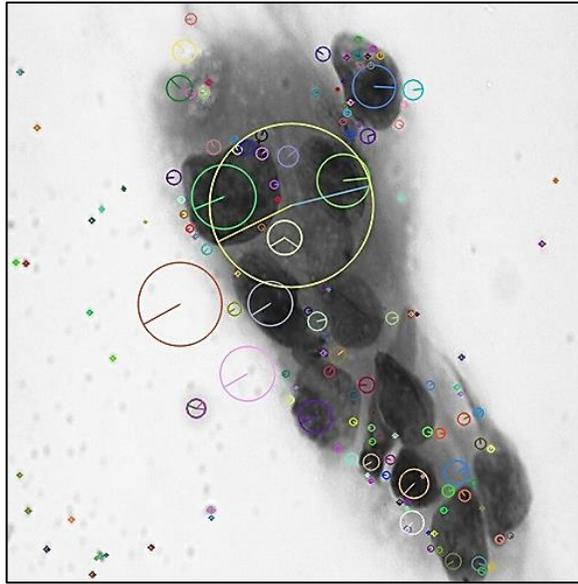


Figura 15. Detección de keypoints con el algoritmo SIFT (fuente: autores).

### 4. Descripción del Keypoint

La representación de cada punto clave, corresponde a un descriptor que contiene parámetros de ubicación, escala y orientación de imagen, obtenidas en las operaciones anteriores. Dicho descriptor proporciona invariancia a estos parámetros.

Teniendo en cuenta la figura 12, cada *Keypoint* tendrá 16 celdas conformadas por la orientación que se dividen en 8 contenedores de 45°. Para el cálculo de las orientaciones y los gradientes del *blob* se lleva a cabo los siguientes pasos:

- Calcular la escala apropiada: correspondiente a la escala del *keypoint*, es decir el índice de variancia estándar,  $\sigma$ , con que fue filtrada la imagen.
- Invariancia de Rotación: las coordenadas del descriptor y las orientaciones del gradiente se rotan con relación a la orientación del punto clave.
- Precálculo de gradientes: se realiza para obtener mayor eficiencia y es determinado por las ecuaciones 12 y 13.
- Asignación de peso: permite evitar cambios repentinos en el descriptor con variaciones de posición y dar menos énfasis a los gradientes lejanos del centro del descriptor. Es realizado con una función gaussiana que tiene un factor de desviación estándar igual a la mitad del ancho de la ventana del descriptor.

Posterior, se crea el histograma para cada una de las 16 muestras, mediante la suma de las magnitudes del gradiente que corresponde a una misma orientación, siendo asignado a uno de los 8 contenedores. Para evitar los efectos de límites, el valor del gradiente se distribuye

en los contenedores adyacentes mediante una interpolación tri-lineal, es decir, cada gradiente se multiplica por un factor de  $1 - d$ , en el cual  $d$  es la distancia de la muestra desde el valor central del contenedor, medida en unidades del espaciado del intervalo del histograma.

Finalmente, la concatenación de los 16 histogramas, determina el descriptor o vector de características para cada *keypoint*, es decir tendrá una dimensión de 128 características. Correspondiente al tamaño del *blob* (4x4 regiones) y el número de orientaciones (8).

### 2.3.1.2 Speeded-Up Robust Features (SURF)

Es un algoritmo compuesto por un detector y descriptor de *keypoints*, basado en SIFT, por lo tanto, los descriptores son invariantes a factores de escala, rotación y parcialmente a iluminación, sin embargo, SURF presenta ventajas en tiempo de procesamiento debido a que utiliza diferentes métodos ligeros y robustos para el cálculo de características locales. El detector encuentra de manera confiable los mismos puntos en diferentes condiciones de perspectiva, cumpliendo con la principal propiedad de un detector, la repetibilidad. Por otra parte, el descriptor es robusto en situaciones de ruido, deformaciones geométricas y errores de detección. La reducida dimensión del descriptor, 64 características, lo hacen ideal para el procesamiento de grandes bases de datos. Finalmente, la evaluación de la coincidencia de los *keypoints* se lleva a cabo con el cálculo de la distancia euclidiana o Mahalanobis [42].

## 1. Detector de Keypoints

El detector de SURF, denominado Fast-Hessian se basa en la matriz Hessiana, debido al buen desempeño en el tiempo de cálculo. A diferencia del detector SIFT que necesita dos medidas para obtener la invariancia en escala (*scale-space*) y en ubicación (Diferencias de gaussianas), SURF, solo utiliza el determinante de la matriz Hessiana para conseguir invariancia en ambos factores. Dado un punto  $p = (x, y)$  en una imagen  $I$ , se define la matriz Hessiana  $H(p, \sigma)$  en  $p$  con una escala  $\sigma$ , expresada por la ecuación 14.

$$H(p, \sigma) = \begin{bmatrix} L_{xx}(p, \sigma) & L_{xy}(p, \sigma) \\ L_{xy}(p, \sigma) & L_{yy}(p, \sigma) \end{bmatrix} \quad (14)$$

Donde  $L_{xx}(p, \sigma)$  es la convolución de la imagen  $I$  en el punto  $p$  con la derivada gaussiana de segundo orden, de igual forma para  $L_{xy}(p, \sigma)$  y  $L_{yy}(p, \sigma)$ . La segunda derivada parcial de la función Gaussiana, denominada *Laplaciana de Gaussiana (LoG siglas en ingles)*, permite

encontrar cambios significativos alrededor del *keypoint*. LoG, es definido por la convolución del operador laplaciano,  $\nabla^2$ , con la función gaussiana  $G(x, y, \sigma)$

$$LoG(x, y) = \nabla^2 * G(x, y, \sigma) \quad (15)$$

Donde  $\nabla^2$  de  $f(x, y)$ , esta definida por la ecuación 16.

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (16)$$

por lo tanto

$$LoG(x, y) = \frac{1}{\pi\sigma^4} \left[ 1 - \frac{x^2 - y^2}{2\sigma^2} \right] e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (17)$$

La utilización del filtro LoG implica alto costo computacional por esta razón, tanto SIFT como SURF presentan aproximaciones a LoG por medio de DoG y filtros de caja, respectivamente. Adicional, para reducir el tiempo de cálculo, SURF, implementa el concepto de imágenes integrales, tal como se define por [56].

Los filtros de caja son una aproximación discretizada y recortada de la gráfica de LoG con un determinado grado de suavización. La Figura 16 ilustra la discretización y aproximación de la primera escala que tiene un grado de suavización de 1.2

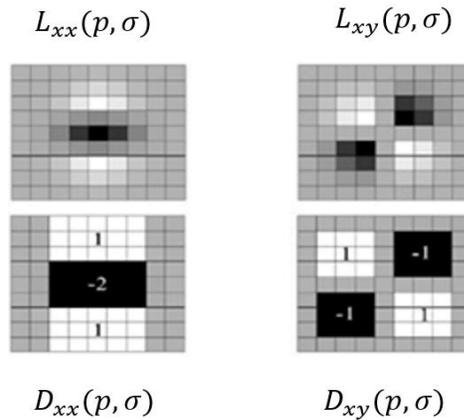


Figura 16. Las cajas superiores representan el laplaciano de forma discretizado, mientras las inferiores son una aproximación a estas (fuente: adaptada de [42]).

Con las aproximaciones expresadas por:  $D_{xx}$ ,  $D_{xy}$  y  $D_{yy}$ , se calcula el determinante a partir de la ecuación 18.

$$\det(H_{aprox}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \quad (18)$$

El factor 0.9 garantiza un equilibrio de los pesos para el determinante y es calculado mediante la norma Frobenius.

El análisis del espacio de escala se lleva a cabo mediante el aumento del tamaño del filtro en lugar de reducir de forma iterativa la dimensión de la imagen, tal como se presenta en SIFT (ver Figura 17).

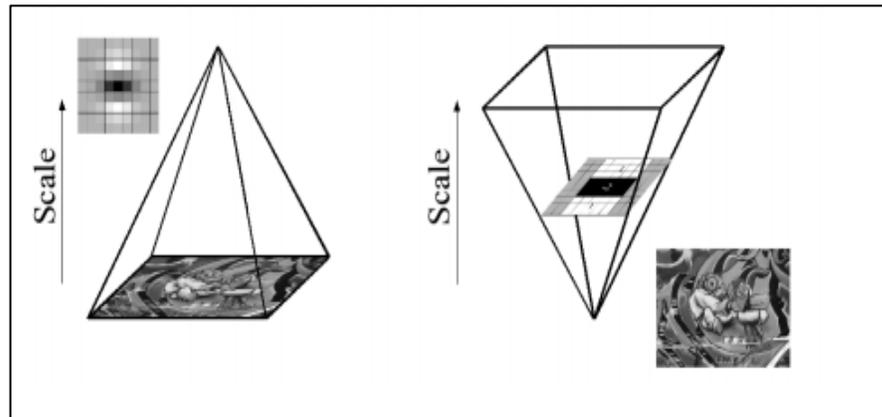


Figura 17. Representación del Space-Scale: SIFT (izquierda), SURF (derecha) (fuente: adaptada de [42]).

La *scale-space* de SURF se divide también en octavas y éstas a su vez en un número fijo de escalas. El factor de incremento de la dimensión de los filtros dentro de cada octava es el doble con relación a la octava anterior, al mismo tiempo la dimensión del filtro inicial en cada octava es el segundo filtro de la octava antecesora. La construcción del espacio escala se inicializa con un filtro de 9x9 con un incremento de 6 píxeles, por lo tanto, las dimensiones de los filtros de la primera octava son: 9x9, 15x15, 21x21, 27x27, etc. Para la segunda octava el factor de crecimiento del filtro será de 12 píxeles con un tamaño inicial de 15x15, dando como resultado dimensiones de 15x15, 27x27, 39x39, 51x51. La Figura 18 da una visión general del crecimiento de los filtros para las tres primeras octavas [42].

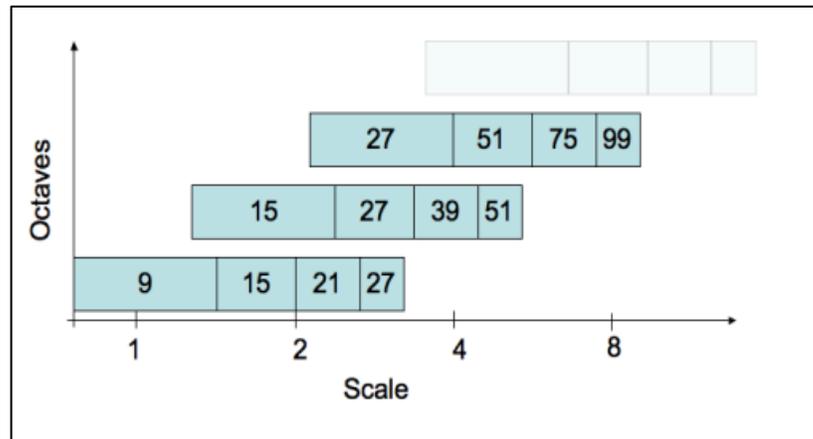


Figura 18. Crecimiento de tamaño de los filtros para las tres primeras octavas (fuente: [42]).

Finalmente, se procede a la localización de los *keypoints* en la imagen y sobre las escalas, en un vecindario de 3x3. De este modo, se tiene en cuenta los máximos del determinante de la matriz Hessiana para ser interpoladas en escala y espacio de la imagen, por medio del método de Brown et al [55].

## 2. Descriptor de Keypoints

La descripción de las características se compone de dos procedimientos, correspondientes a la asignación de orientación, con un análisis del contenido local por medio de una circunferencia alrededor del punto de interés y la descripción de las características a través de un área rectangular con orientación calculada [42].

### 2.1 Asignación de Orientación

La identificación de una orientación reproducible para los *keypoints*, permite la invariancia a factores de rotación de la imagen. Como primera medida se obtiene las respuestas de los filtros Haar-Wavelet en direcciones  $x$  e  $y$  de la imagen integral mediante un círculo de radio  $6s$ , siendo  $s$  la escala donde se detectó el *keypoint*. La Figura 19 muestra los filtros binarios Haar-wavelet con su correspondiente respuesta.



Figura 19. Respuestas de filtros Haar-Wavelet (fuente: adaptada de [42]).

Una vez obtenidas las respuestas Haar-Wavelet son ponderadas con una función gaussiana de segundo orden ( $\sigma = 2.5s$ ). Dichas respuestas son representadas en un sistema de coordenadas, centrado en el *keypoint*. Finalmente, la orientación dominante es determinada por el vector más grande, obtenido de la suma de todas las respuestas dentro de un segmento circular de  $60^\circ$  que se desplaza. La Figura 20 muestra las respuestas con puntos azules y la correspondiente orientación ilustrado con una flecha roja.

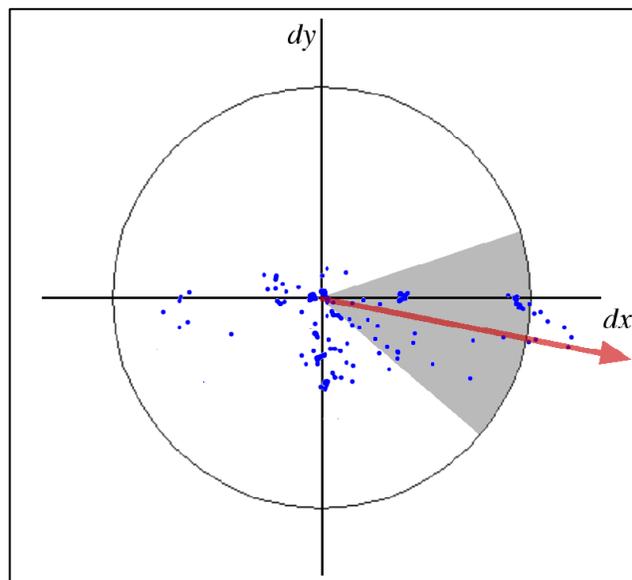


Figura 20 Asignación de orientación (fuente: adaptada de [42]).

## 2.2 Descripción de Características

Con una región cuadrada de dimensión  $4 \times 4$  (ver Figura 21), denominado *blob*, centrado en el *keypoint* y orientado en la dirección estimada en la sección anterior, se calcula el descriptor. El tamaño de la región corresponde a  $20s$ .

Las respuestas de Haar-Wavelet (tamaño  $2s$ ) en dirección horizontal y vertical se denota  $dx$  y  $dy$  respectivamente. Para garantizar mayor robustez de las respuestas,  $dx$  y  $dy$ , ante deformaciones geométricas y errores de posición, éstas son ponderadas con una función gaussiana de variancia estándar igual a  $3.3s$  ( $\sigma = 3.3s$ ). Con la sumatoria de las respuestas horizontales y verticales en cada sub-región del *blob*  $4 \times 4$ , se forma un primer conjunto de valores para el vector de características. Con el fin de conocer la polaridad sobre los cambios de intensidad se calcula la suma de los valores absolutos de las respuestas,  $|dx|$  y  $|dy|$ . Por lo tanto, cada sub-región es representada por un vector de cuatro dimensiones,  $v = (\sum dx, \sum dy, \sum |dx|, \sum |dy|)$ , obteniendo un descriptor de 64 características para cada punto clave.

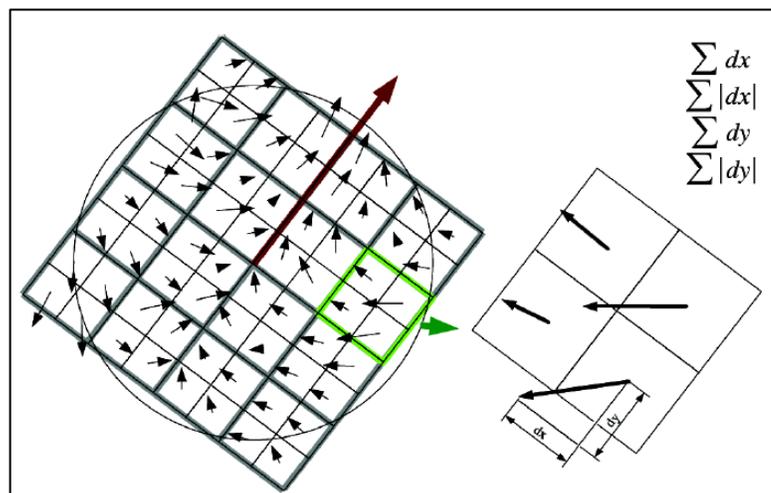


Figura 21. Cuadrícula  $4 \times 4$  sub-regiones (izquierda). Cada sub-divisiónes  $2 \times 2$  se calcula  $dx$ ,  $dy$ ,  $|dx|$  y  $|dy|$  (fuente: [42]).

### 2.4.1.3 OpponentSIFT y OpponentSURF

La principal desventaja de SIFT y SURF radica en omitir los cambios de color, ya que solo utilizan intensidades de escalas de grises. Para suplir esta debilidad, existen dos enfoques en BoF con el fin de crear un *codebook* de información de color (descriptor OpponentSIFT) y forma (descriptor SIFT), sin embargo, se puede extender para incluir otras variables. En la primera, implica fusionar descriptores de color con los descriptores de forma, de esta manera se aumenta el poder discriminativo de las características locales. Esta técnica es llamada

*early-fusion*. En la segunda, definida como *late-fusion*, concatena el histograma de color y forma, obtenidas de manera independiente [57] [58].

Los descriptores de color se fundamentan en la técnica *early-fusion*, debido a la combinación de tres histogramas, uno por cada color primario del espacio de color RGB, de igual forma, existen descriptores derivados de otros espacios de color, tal es el caso de OpponentSIFT, donde transforma el espacio RGB a los canales del espacio de color del oponente,  $(O_1, O_2, O_3)$ , utilizando la ecuación 19.

$$(O_1 \ O_2 \ O_3) = \left( \frac{R - G}{\sqrt{2}}, \frac{R + G - 2B}{\sqrt{6}}, \frac{R + G + B}{\sqrt{3}} \right) \quad (19)$$

El canal  $O_3$ , representa la información de intensidad, mientras que los canales  $O_1$  y  $O_2$  contienen la información de color de la imagen. Finalmente, el descriptor OpponentSIFT caracterizará la información por cada canal, por lo tanto la dimensión vectorial de un *keypoint* será de 384 características [59]. El estudio del desempeño de los descriptores de color derivados de SIFT realizado por Van de Sande et al [60] muestran un buen rendimiento de OpponetSIFT en el reconocimiento de objetos y escenas. De igual forma se establece la anterior información para el descriptor OpponentSURF, con una dimensión del vector igual 192 características.

### 2.3.2 Construcción del Codebook

Una vez se caracteriza la información visual de cada imagen, se crea un vocabulario visual (*Codebook*) de los vectores de características. Para ello, se lleva una cuantificación vectorial con métodos de *Clustering* los cuales permiten la división de los descriptores en ciertos grupos (*clústeres*), correspondiente a las palabras visuales (*Codewords*). La creación del *codebook* permite reducir la dimensionalidad de los datos de entrada ya que se obtiene  $K$  palabras visuales representando cada grupo de características y descripciones similares. Los grupos de palabras visuales garantizan la construcción del histograma representativo, el cual es la entrada para el entrenamiento del algoritmo de clasificación.

En la implementación, el algoritmo basado en el error cuadrático más simple y más utilizado en *Bag of Features* para la creación de *codebooks* es *K-means*, debido a que este algoritmo representa cada agrupamiento de descriptores mediante un vector, siendo este tipo de entrada más adecuada para el proceso de clasificación en BoF. Este algoritmo de aprendizaje no supervisado sigue una forma simple de clasificar un conjunto de datos a través de un cierto número  $K$  de clústeres, el cual será representado por un centroide. El algoritmo asume que se forma un espacio vectorial a partir de las características de los datos y trata de identificar el agrupamiento [61]. Sea  $X = \{x_i, i = 1, 2, \dots, N\}$  los puntos  $N$ -dimensionales que se agrupan

en un conjunto de  $K$  grupos,  $C = \{C_i, i = 1, 2, \dots, k\}$ . El algoritmo *K-means* encontrará particiones tales que el error cuadrado entre la media empírica de un clúster  $C_j$  y los puntos  $x_i$ , para este mismo clúster sea minimizado. Finalmente, el objetivo de *K-means* será minimizar la suma del error al cuadrado sobre todos los  $K$  clústeres, los cuales se describen por la siguiente ecuación:

$$\min_{u_1 \dots u_k} E(C_k) = \sum_{j=1}^k \sum_{i=1}^n \|x_i^{(j)} - c_j\|^2 \quad (20)$$

Donde  $\|x_i^{(j)} - c_j\|^2$  es la distancia euclidiana que mide la similitud entre las características de sus respectivos centroides.

A continuación, se describe el procedimiento llevado a cabo por el algoritmo

1. Coloca de manera aleatoria los  $K$  puntos de cada clúster del conjunto de entrenamiento  $\{C_1, C_2, \dots, C_k\}$ . Estos puntos representan los centroides iniciales de los clústeres.
2. Para cada descriptor se calcula la distancia euclidiana con respecto a los centroides
3. Asigna cada descriptor al clúster que tiene el centroide más cercano.
4. Cuando se hayan asignado todos los descriptores, vuelve a calcular las posiciones de los centroides  $K$ .
5. Repite los pasos 3 y 4 hasta que los centroides ya no se muevan. Esto produce una separación de los descriptores en grupos a partir de los cuales se puede calcular la métrica que se va a minimizar.

En la Figura 22 se muestra la interacción de los anteriores pasos para la división de una población de 800 individuos en 5 grupos, mediante *K-means*.

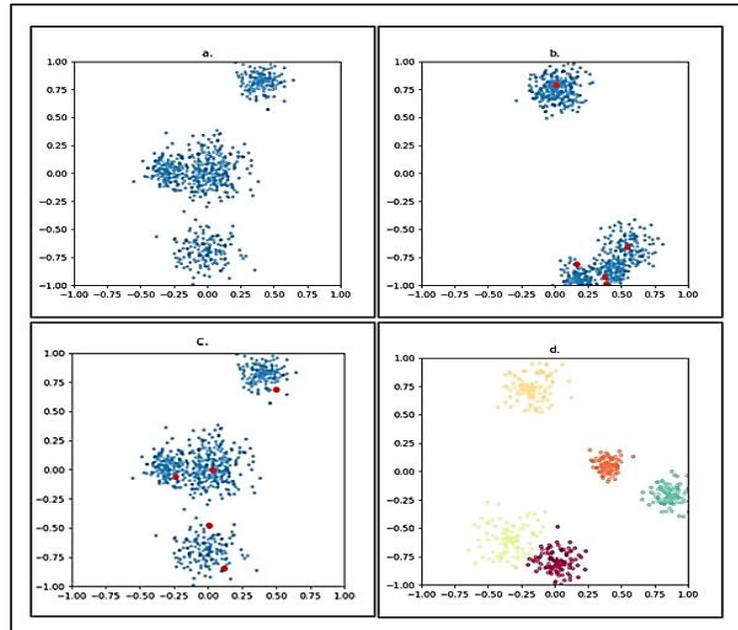


Figura 22. Interacción de las fases del algoritmo *K-means* (fuente: autores).

A pesar de que el algoritmo *K-means* es robusto, no siempre encuentra la configuración más óptima, que corresponde al mínimo de la función objetivo, definida por error al cuadrado. Para reducir este efecto se ejecuta varias veces. Además, es sensible al número de centroides iniciales definidos manualmente. Parámetro importante en la construcción del vocabulario ya que delimita el tamaño de este. De acuerdo con diferentes trabajos en clasificación de imágenes médicas, un vocabulario de gran tamaño es lo más adecuado [10], sin embargo, Tomassi et. al [21] demostró que el tamaño del diccionario visual no es un aspecto significativo en la tarea de clasificación de imágenes de radiología. Por esta razón, es necesario evaluar diferentes tamaños de vocabularios visuales para analizar el desempeño de clasificación.

### 2.3.3 Representación Bag of Features (BoF)

La representación de las imágenes se lleva a cabo cuando se compara los *Keypoints* obtenidos con las palabras visuales del *codebook*. De esta forma se logra cuantificar los parches en un histograma de ocurrencias mediante el criterio de similitud, obteniendo el vector de frecuencia, denominado vector final. Sin embargo, el número de descriptores por cada imagen es aleatorio, siendo un problema a la hora de evaluar la comparación, por lo cual es necesario normalizar. Esto es pertinente cuando el método de extracción de características es

de puntos de interés[8]. Por esta razón, se presenta a continuación una de las técnicas de normalización más comunes del método BoW y BoF.

### 2.3.3.1 Term Frequency-Inverse Document Frequency

Es la técnica más común, para analizar problemas de clasificación de archivos de textos, por medio del peso o importancia de la cuantificación de los vectores de características, logrando representar los documentos para agruparlos en una categoría [62]. Esto es una estrategia de mitigación para los problemas de cuantificación que ocurren cuando los descriptores se distribuyen de tal manera que las estrategias de agrupamiento sobre-producen descriptores y subestiman otros [63]. Los pesos o importancia asignada a cada término, permite penalizar los descriptores más comunes y enfatizar en aquellos que son más únicos. Este es el objetivo del esquema de “*Term Frequency – Inverse Document Frequency*” (TF-IDF), donde el valor a cada descriptor en el histograma está definido por dos valores, y su representación se muestra en la ecuación 21.

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right) \quad (21)$$

Donde  $tf_{i,j}$  determina a partir del número de veces que el descriptor  $i$  aparece en la imagen  $j$  ( $n_i^j$ ). Para evitar inconsistencias el término,  $tf_{i,j}$ , se normaliza por la cantidad total de descriptores,  $N_j$ , de la imagen en cuestión (22). La ecuación 21 define  $tf$ .

$$tf_{i,j} = \frac{n_i^j}{N_j} \quad (22)$$

El segundo término,  $\log\left(\frac{N}{df_i}\right)$ , identificado como “documento inverso de frecuencia”, IDF (siglas en ingles), sirve para penalizar los pesos de los términos comunes. Si el cociente entre el número total de descriptores,  $N$ , y el número de imágenes donde aparece el descriptor  $i$  ( $df_i$ ), tiende a un valor grande, el peso asignado a dicho descriptor será bajo y viceversa [49].

### 2.3.4 Clasificación

Una vez se tiene el vector global de representación de las imágenes, el siguiente paso es clasificar las imágenes en las siete clases del sistema Bethesda 2001. Primero, es necesario

crear un modelo o descubrir patrones en el conjunto de descriptores, esto permite discriminar características de las clases separadas. Para ello, se pueden utilizar diferentes métodos de clasificación automático, sin embargo, los algoritmos más utilizados se denominan paramétricos. Estos requieren de un entrenamiento previo donde obtienen el modelo discriminativo. Dentro de los clasificadores paramétricos se destaca las máquinas de soporte vectorial (SVM), Redes Neuronales Artificiales (ANN),  $K$  vecinos más cercanos (KNN), etc. En principio se puede utilizar cualquier clasificador para resolver el problema de separar imágenes, no obstante, en el campo de imágenes histopatológicas se destaca SVM, explicado en la siguiente sección.

### 2.3.4.1 Máquinas de Soporte Vectorial

Este tipo de algoritmo fue introducido por Vapnik et al [64], el cual se basa en la utilización de un espacio de separación de clases, denominado *hiperplano*. Aunque en un principio se creó para problemas de clasificación binaria, es decir dos clases, actualmente se utilizan en enfoques multiclases caracterizados por ser de muy alta dimensión, tal como se presenta en la información descrita de las imágenes.

Para entender cómo funcionan las SVM, se analiza el caso cuando el conjunto de entradas es linealmente separable. En un principio, se realiza un proceso de aprendizaje en el cual permite determinar el mejor hiperplano para clasificar el conjunto de datos, dependiendo de la clase a la que pertenece. Para ello se tiene un conjunto binario de muestras en un espacio bidimensional,  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , donde  $x_i \in \mathbb{R}^d$  e  $y_i \in \{+1, -1\}$ . El conjunto  $S$  conformado por muestras en las clases  $\{+1, -1\}$ , serán separadas sin dificultad alguna a través de una función lineal, obteniendo la siguiente frontera de decisión

$$w * x_i + b = 0 \quad (23)$$

Donde  $w$  es el vector perpendicular al hiperplano,  $x_i$  es el espacio de características y  $b$  es la constante del punto de corte con el eje de la ordenada. La solución del conjunto de datos de entrenamiento obedece a la siguiente condición de clasificación:

$$\begin{aligned} w * x_i + b &\geq 0 & \text{si } y_i &= +1 \\ w * x_i + b &\leq 0 & \text{si } y_i &= -1, \quad i = 1, \dots, n \end{aligned} \quad (24)$$

Por otra parte, las muestras que cumplen con la igualdad en (24) determinan un par de vectores,  $h^+$  y  $h^-$ , denominados vectores de soporte, de aquí proviene el nombre del algoritmo.

$$\begin{aligned}h^+ &\rightarrow w * x_i + b = +1 \\h^- &\rightarrow w * x_i + b = -1\end{aligned}\tag{25}$$

La distancia entre una muestra y el hiperplano óptimo es conocido como *margen*, denotado con  $d$ . Cuando  $d$  es máximo cumple la condición de *margen óptimo*, el cual tiene la propiedad de ser equidista con la muestra más cercana de cada clase.

$$d^+ = d^- = \frac{|w * x_i + b|}{\|w\|}\tag{26}$$

La búsqueda del hiperplano óptimo implica encontrar el valor de  $w$ , el cual maximiza el margen. Sin embargo, existe una familia de soluciones lineales  $\delta(w * x_i + b)$ , representando el mismo hiperplano. Para establecer una sola solución, se considera la anterior expresión y se asigna de forma arbitraria la unidad al término  $\|w\|d$ , es decir

$$d^{+/-} \|w\| = 1\tag{27}$$

Por lo tanto, el margen es inversamente proporcional a la norma de  $w$ . La muestra que cumple la siguiente ecuación, significa que pertenece al conjunto de vectores de soporte

$$d^{+/-} = \frac{1}{\|w\|}\tag{28}$$

La Figura 23 ilustra los anteriores conceptos.

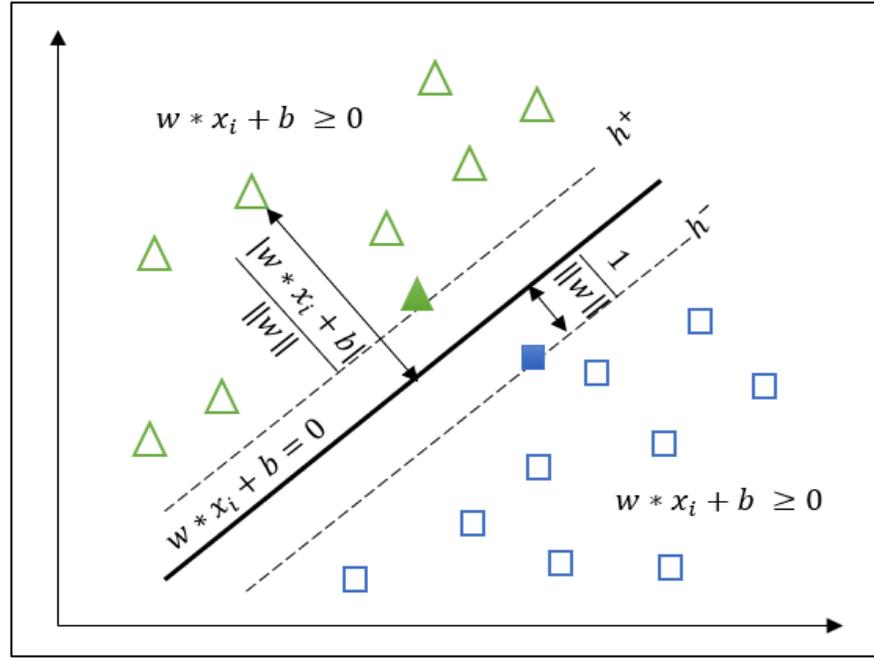


Figura 23. Separación de dos clases en caso linealmente separable (fuente: autores).

Encontrar el margen máximo entre las dos proyecciones de los datos de entrenamiento de las clases  $\{+1, -1\}$  se puede generalizar como la búsqueda de los valores  $w$  y  $b$  con el fin de minimizar la función  $f(w) = \|w\|$ , es decir:

$$\min f(w) = \frac{1}{2} \|w\|^2 \quad (2919)$$

La teoría de la optimización permite resolver el problema cuadrático de la anterior expresión. Todo problema de optimización, denominado primal, tiene una forma dual, solo si la expresión en cuestión es una función convexa. Debido a que la mayoría del espacio de características que presenta las imágenes es de muy alta dimensionalidad, es conveniente transformar a la forma dual ya que permite reducir la resolución. El procedimiento para resolver el problema dual se basa en la función lagrangiana y se detalla en [65], obteniendo la expresión 30:

$$\max \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \quad (30)$$

Si se tienen en cuenta  $S$  como un conjunto no linealmente separable, la definición anterior se puede generalizar introduciendo un conjunto de variables reales positivas, denominadas variables de holgura,  $\xi_i$ , por lo tanto, la expresión (25) queda modificada a (31).

$$y_i (w * x_i + b) \geq 1 - \xi_i \quad (31)$$

Para un par de datos  $(x_i, y_i)$ ,  $\xi_i$  representa la desviación del caso separable. Entonces la suma del conjunto de variables de holgura es una medida de error en la clasificación, redefiniendo el cálculo del hiperplano óptimo (expresión 32).

$$\begin{aligned} \min f(w, \xi_i) &= \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{sujeto a } y_i (< w, x_i > + b) + \xi_i - 1 &\geq 0 \\ \xi_i &\geq 0, \quad i = 1, \dots, n \end{aligned} \quad (32)$$

Donde  $C$  es una constante, definida por el usuario. El valor de  $C$  ajusta un balance entre la maximización del margen y la penalización de clasificación. De este modo, un valor grande implica que el SVM sea más estricto, es decir, entrega valores mínimos de  $\xi_i$ . En caso contrario, un índice de  $C$  inferior, resultará con valores muy grandes de desviación, que en pocas palabras indicaría un número elevado de muestras mal clasificadas.

La solución es calculada por la función *lagrangiana*, tal como se procede en el caso linealmente separable (expresión 33)

$$\begin{aligned} \max W(\alpha) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j < x_i, x_j > \\ \text{sujeto a } \sum_{i=1}^n \alpha_i y_i &= 0 \\ 0 \leq \alpha_i &\leq C, i = 1, \dots, n \end{aligned} \quad (33)$$

La anterior expresión determina el espacio óptimo de separación en términos de  $\alpha$ . Cuando  $\alpha = C$ , corresponde a problemas no separables, en estos casos generalmente se transforma el espacio de entradas originales en un nuevo espacio de mayor dimensión, denominado espacio de *Características* o *Hilbert*, por medio de una función,  $\varphi$ , es decir  $\varphi : \mathbb{X} \rightarrow \mathbb{F}$ , donde  $\varphi(x) = [\varphi_1(x), \dots, \varphi_n(x)]$  tal que  $\varphi_i(x)$  es una función no lineal. Por lo tanto, el fin es determinar un hiperplano en el espacio de características  $\mathbb{F}$ . La Figura 24 ilustra la transformación al espacio de características.

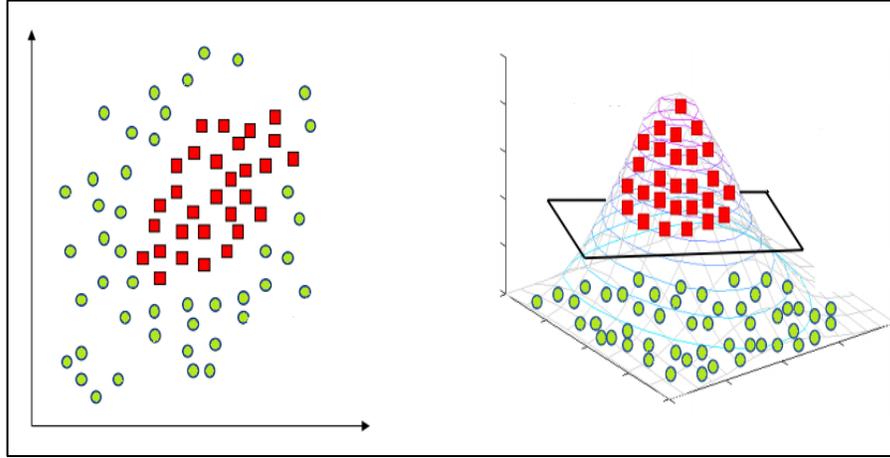


Figura 24. Transformación de espacio original a espacio de características a través de la función kernel (fuente: adaptada de [66]).

Dicha transformación se logra calculando el producto interno de los datos de entrada,  $\langle x_i, x_j \rangle$ , es decir,

$$x_i \cdot x_j = \varphi(x_i) \cdot \varphi(x_j) = K(x_i, x_j) \quad (34)$$

Donde  $K(x_i, x_j)$  es denominado *Kernel*. El cálculo del hiperplano óptimo cumple con resolver el mismo problema dual de la expresión (30) bajo el espacio de características, es decir (35)

$$\begin{aligned} \max W(\alpha) &= \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{sujeto a } &\sum_{i=1}^n \alpha_i y_i = 0 \\ &0 \leq \alpha_i \leq C, i = 1, \dots, n \end{aligned} \quad (35)$$

La elección de la función es un problema en esta estrategia, ya que debe cumplir con transformar los datos en un espacio separable de mayor dimensión. la solución está dada por el cumplimiento de las condiciones del teorema de *Mercer* [67] ya que pueden ser usadas como productos escalares. Las funciones *kernels* más comunes en la implementación como núcleo se nombran a continuación:

- Lineal:  $K(x_i, x_j) = \langle x_i, x_j \rangle \quad (36)$

- Polinomial:  $K(x_i, x_j) = [\gamma \langle x_i, x_j \rangle + \tau]^p \quad (37)$

- Base Radial:  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$  (38)
- Sigmoidal:  $K(x_i, x_j) = \tanh(\gamma \langle x_i, x_j \rangle + \tau)$  (39)

Donde  $\gamma$ ,  $\tau$ , y  $p$  son hiper-parámetros de los *kernels*. Sus valores determinan el desempeño de clasificación, conllevando en el estudio y desarrollo de diferentes metodologías de sintonización del modelo SVM.

### 2.3.4.2 Búsqueda de Hiper-parámetros

Los algoritmos de aprendizaje deben ser ajustados para encontrar un rendimiento óptimo de la tarea a ejecutar, es decir, ajustan los hiper-parámetros para generalizar la información de las diferentes clases. Esto se lleva a cabo entrenando el algoritmo con un conjunto combinado e independiente de hiper-parámetros, donde el promedio de la función de pérdida determina el modelo final. Generalmente el rendimiento de generalización es estimado por medio de la *Validación Cruzada* [67]. Este tipo de técnica divide aleatoriamente una muestra de datos en  $k$  submuestras de igual tamaño. Una de estas submuestras es utilizada para realizar la validación del modelo, y las  $k - 1$  submuestras restantes son aplicadas para entrenar el algoritmo. Finalmente, el promedio de los  $K$  errores de clasificación es el error de validación cruzada [68].

$$CV(K) = \frac{1}{K} \sum_{l=1}^K (y_l - \hat{y}_l)^2 \quad (40)$$

Un método común en el cálculo de los hiper-parámetros es la *búsqueda de grilla o barrido de parámetros*, determinado por el tipo de kernel. Considerando un kernel RBF, por lo tanto, el objetivo de este método es identificar el par de  $(C, \gamma)$ , donde  $C$  es el parámetro común de las SVMs de penalización y  $\gamma$  es el parámetro del núcleo RBF. Para cada parámetro se define una secuencia de valores generalmente con incremento logarítmico [67].

$$\begin{aligned} \gamma &= 2^{-15}, 2^{-13}, \dots, 2^3 \\ C &= 2^{-5}, 2^{-3}, \dots, 2^{15} \end{aligned}$$

Las diferentes combinaciones se establecen en una cuadrícula, donde por cada par se calcula el error de validación cruzada del conjunto de iteraciones o las  $k$  submuestras (ver Figura 25).

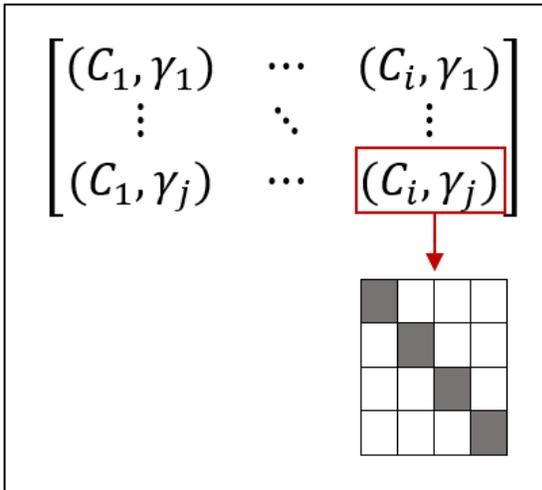


Figura 25. Esquema de búsqueda grilla y validación cruzada (fuente: adaptada de [67]).

Cada fila corresponde a una iteración, donde las divisiones grises representan el conjunto de validación y las blancas los datos de entrenamiento. Finalmente, el par de hiper-parámetros con el menor error serán retornados, definiendo así el modelo SVM [67].

### 2.3.4.3 SVM Multi-clases

La naturaleza del SVM es calcular un hiperplano óptimo en el espacio de características para separar dos clases. Este concepto se ha adaptado para realizar estrategias que dividen el problema original en múltiples problemas binarios con el fin de extender las SVM a K-clases. La mayoría de estas estrategias están incluidas en el marco *Error-Correcting Output Codes* (ECOC)[69], sin embargo, los esquemas más comunes son: *one-versus-one* (OVO) y *one-versus-all* (OVA). Detallados a continuación

- *One-vs-One*

En esta estrategia se obtiene la misma cantidad de subproblemas como pares de clases podemos obtener, es decir consiste en entrenar un clasificador para cada par de clases. Transforma un problema K-clases en  $K(K - 1)/2$  clasificadores binarios. Por ejemplo, un SVM binario podría comparar la clase  $K_i$ , entrenado con un conjunto de etiqueta  $i$ , y la clase  $K_j$ , con un conjunto de etiqueta  $j$ , mientras que las muestras de las clases restantes son ignoradas. Finalmente, la clasificación final se realiza asignando la observación de la prueba a la clase a la que se asignó con mayor frecuencia en estas clasificaciones por pares.

- *One-vs-All*

Este tipo de estrategia define la comparación de una clase  $k$  para discriminar con las clases restantes. Transforma un problema de  $k$  clases en  $k$  problemas binarios, de manera que, la clase  $K_i$  entrenada con muestras de etiqueta  $i$ , es comparada con las restantes  $K - 1$  clases de etiqueta  $j$ .

En estos métodos, la salida deseada,  $y_n$ , del conjunto de entrenamiento,  $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ , es definida por un matriz codificada  $M_{(T,K)}$ , donde  $T$  es el número de clasificadores binarios y  $K$  el número de clases, de esta forma el algoritmo de aprendizaje es entrenado con conjunto pares,  $(x_n, M_{(T,K)})$ , donde  $y_n$  es el conjunto de etiquetas de las clases  $K$ , por lo tanto,  $(x_n, M_{(T,y_n)})$ . La matriz  $M$  para las estrategias OVO y OVA, se ilustra en la Figura 26.

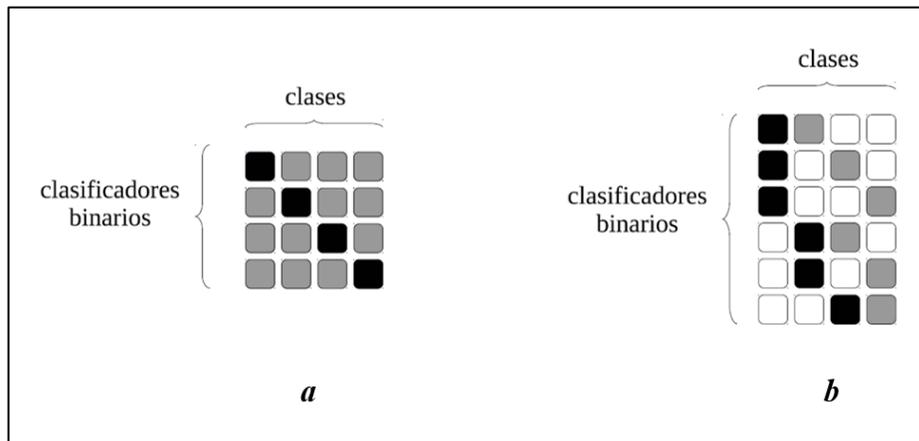


Figura 26. Comparación de multiclasificación con: **a.** OVA y **b.** OVO (fuente: adaptada de [69]).

En la Figura 26 (a) los cuadros negros representan la clase “One” y los cuadros grises “All”, mientras que en la Figura 26 (b) los cuadros negros y grises representan las clases “One” y los cuadros blancos representan las muestras de etiquetas de las clases omitidas [69].

# Capítulo 3

## Clasificación de Imágenes Citológicas.

---

En este capítulo se presenta en detalle el desarrollo de los algoritmos que conforman el sistema de clasificación BoF en imágenes citológicas del estándar Bethesda 2001. Se presenta de forma general en la sección **3.1** las características con las cuales se basan los patólogos para la clasificación de las imágenes bajo el sistema Bethesda 2001. En la sección **3.2** se especifica el modus operandi con el que se creó el banco de imágenes teniendo en cuenta las clases descritas anteriormente. En la sección de **3.3** se describe el esquema del sistema propuesto, abarcando desde la etapa de entrenamiento hasta las métricas de rendimiento del modelo de clasificación.

La implementación del sistema propuesto se desarrolló en el lenguaje de programación *Python*, utilizando la librería especializada de procesamiento de imágenes, *OpenCV*, en conjunto con la librería *numpy*, para el manejo matricial y *Scikit-Learn* para las funciones de aprendizaje automático. Finalmente, para el desarrollo de la interfaz gráfica se utilizó la plataforma de *PyQt4*. La instalación de las herramientas y librerías se exponen en el **Capítulo A** de los anexos, igualmente el manual de uso del aplicativo.

### 3.1 Bethesda 2001

El sistema Bethesda (BTS) 2001 fue elaborado para clasificar muestras citológicas cervicales, de tipo convencional y líquida. Este sistema estandarizado de clasificación es la evolución de la interpretación de las células en superficies epiteliales. Todo inicia en 1925, cuando el médico Papanicolaou descubre las células naviculares, responsables de la gestación debido a la alta densidad de glucosa. En 1941, publica la identificación de células cancerosas del cuello uterino en extendidos vaginales de pacientes fuera de sospecha de cáncer, dos años más tarde presenta la monografía “*Diagnóstico del cáncer uterino por el extendido vaginal*” dando inicio al examen de la citología o papanicolaou. De esta forma James Reagen establece en 1958 el término displasias, clasificadas en leve, moderada y severa, para introducir una terminología estándar del reporte final. Sin embargo, en 1967, Richard plantea uno de los términos más utilizados en la actualidad, NIC (*neoplasia escamosa intraepitelial*), enumerados con I, II y III, e igualando el NIC I con displasia leve, NIC II con displasia moderada, NIC III con displasia severa y el carcinoma in situ como el grado de malignidad

más alto. En 1988, el Instituto Nacional del Cáncer de Estados Unidos promueve el “Sistema Bethesda” con el calificativo SIL (*squamous intraepithelial lesions*). Posteriormente, en 1991 y en mayo de 2001, se concluyó el sistema estandarizado de clasificación de malignidad celular citológico, Bethesda 2001, constituido por seis clases, las cuales se detallan a continuación y son definidas en [26].

- **Normales:** Según el sistema Bethesda, en esta clase se reportan las muestras sin anomalías celulares, es decir, que las muestras que son clasificadas como negativas de lesión o negativas de cáncer intraepitelial. Aunque también puede que se presenten infecciones comunes o de inflamación, pero que no están asociadas al cáncer. Los criterios morfológicos son descritos en la sección 2.1 donde se definió las diferentes células que conforman el epitelio escamoso no queratinizado.
- **ASCUS (células escamosas atípicas de significado indeterminado):** clase de anormalidad conocida como “células escamosas atípicas de significado indeterminado”. Su significado de malignidad indica leves cambios no completamente normales en las células del epitelio escamoso siendo cuantitativa o cualitativamente poco probable para una lesión intraepitelial escamoso (ver Figura 27). Los criterios morfológicos en esta clase son los siguientes:
  - Los núcleos tienen aproximadamente entre dos veces y media y tres veces el tamaño del área del núcleo de una célula normal es decir alrededor de  $35\mu m^2$ .
  - Existe un leve aumento de la relación existente entre el área nuclear y el área citoplasmática (N:C) a comparación de la normal.
  - Hay una hiperchromasia nuclear mínima y se presenta una irregularidad de la distribución morfológica del núcleo.

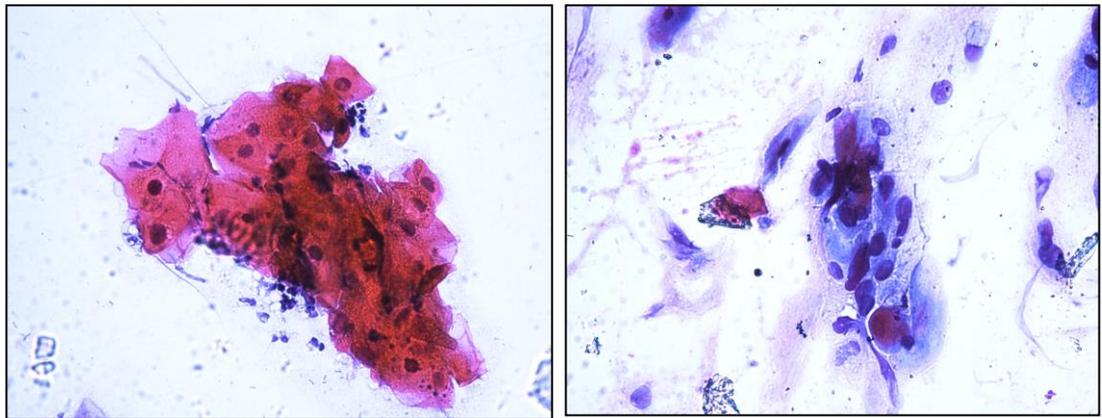


Figura 27. Representación morfológica de la clase ASC-US (fuente: autores).

- **LSIL (lesión intraepitelial escamosa de bajo grado):** clase asociada por los cambios celulares debidos al virus del papiloma humano (HPV). Las células tienen un tamaño nuclear que supera el triple del tamaño del área del núcleo de una célula intermedia, tal como se muestra la Figura 28.

Los criterios morfológicos para esta clase son los siguientes:

- Las células se encuentran aisladas.
- Las células se visualizan con un tamaño generalmente grande y el citoplasma “maduro” o superficial es bastante abundante y bien definido.
- El núcleo de la célula supera el triple del tamaño normal de una célula intermedia normal, es decir hay un leve aumento de la relación (N:C).
- La hiper cromasia nuclear presenta diferentes grados de variación acompañada de binucleación y multinucleación.

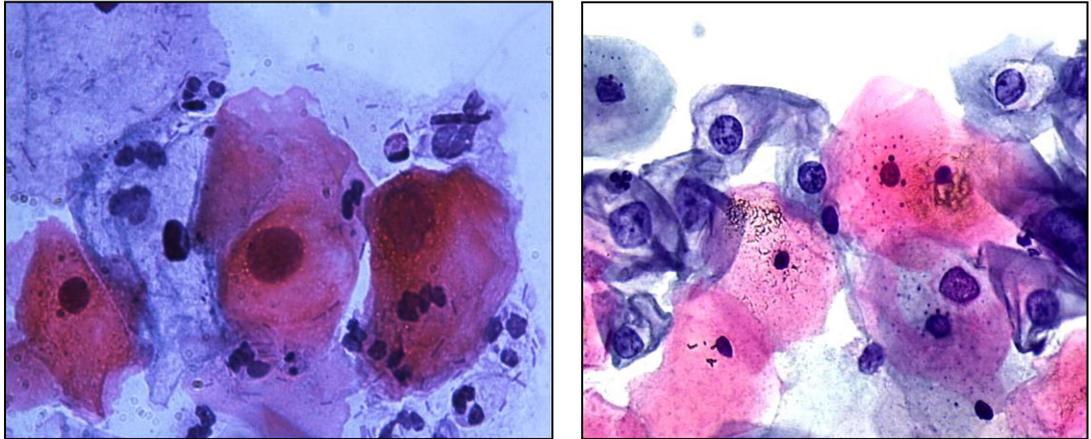


Figura 28. Representación morfológica de la clase L-SIL (fuente: autores).

- **ASCH** (células escamosas atípicas de alto grado): las células se caracterizan por presentar cambios moderados a severos. En esta clase no se puede descartar la posibilidad de que pueda presentarse una lesión intraepitelial escamosa de alto grado (HSIL), la cual es una clase de mayor malignidad.

Los criterios morfológicos para esta clase son (ver Figura 29):

- Las células tienen el mismo tamaño que las células metaplásicas y un núcleo que es entre 1 y 2 veces más grande de lo normal.
- Las células se pueden encontrar de manera aislada formando pequeños grupos de menos de 10 células, que a veces se extienden en forma de “hilera” dentro de la muestra.

- La relación N:C puede ser similar a la de HSIL.

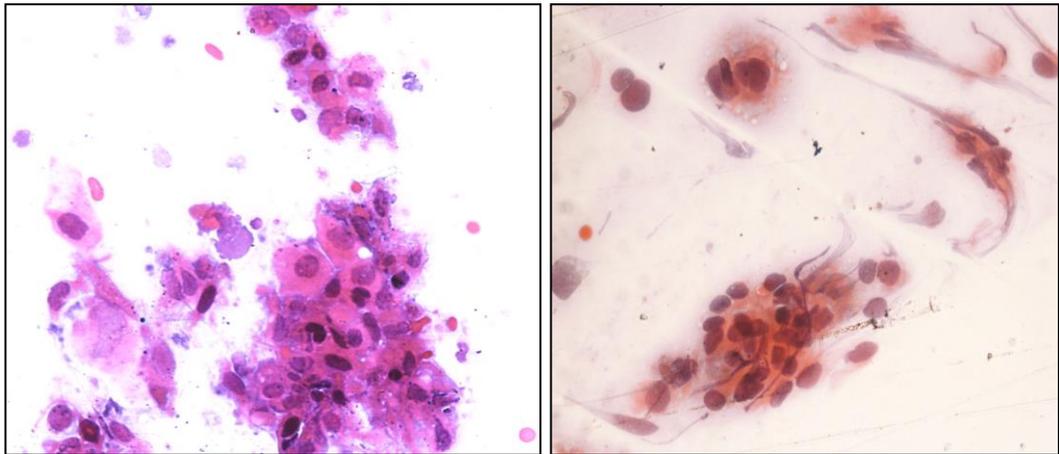


Figura 29. Representación morfológica de la clase ASC-H (fuente: autores).

- **HSIL** (lesión intraepitelial escamosa de alto grado): En esta clase se presenta un alto grado de posibilidad de que las anomalías en las células se puedan convertir en cáncer si no se da un tratamiento preventivo. El alto grado significa que hay cambios más marcados en el tamaño y en la forma de las células anormales y que las células difieren en gran medida de las normales (ver Figura 30).

Los criterios morfológicos para esta clase son:

- Las células se pueden encontrar de forma aislada y en forma de láminas.
- El tamaño de las células es variable, ya que se pueden presentar tamaños parecidos a lesiones de bajo grado hasta el tamaño bastante pequeño como las células de tipo basal.
- La tinción del núcleo y la hipertrofia es más variable que en las lesiones de bajo grado y está acompañado de variaciones de tamaño.
- El agrandamiento del núcleo es parecido al de las lesiones de bajo grado, pero a diferencia de estas presentan un citoplasma reducido, es decir la relación (N:C) es mayor.
- En cuanto al contorno de la membrana nuclear, esta presenta bastantes irregularidades.

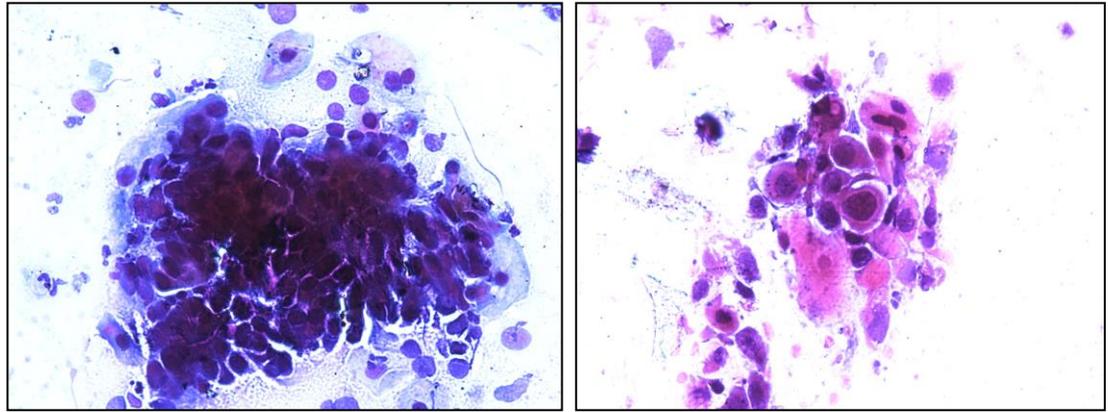


Figura 30. Representación morfológica de la clase H-SIL (fuente: autores).

- **Carcinoma:** En esta clase las células escamosas anormales han invadido profundamente tejidos y órganos. En otras palabras, se da esta clasificación cuando está presente el cáncer del cuello uterino.

En cuanto a los criterios morfológicos para esta clase son:

- Los núcleos presentan gran variabilidad de tamaño y las membranas nucleares pueden poseer también configuración irregular, y además de ello se suele encontrar numerosos núcleos hiper cromáticos de gran intensidad.
- Las células suelen estar aisladas, con un número bajo de población y con menos frecuencia en conglomerados.
- Alrededor de las células suelen estar acompañadas por glóbulos rojos (ver Figura 31).

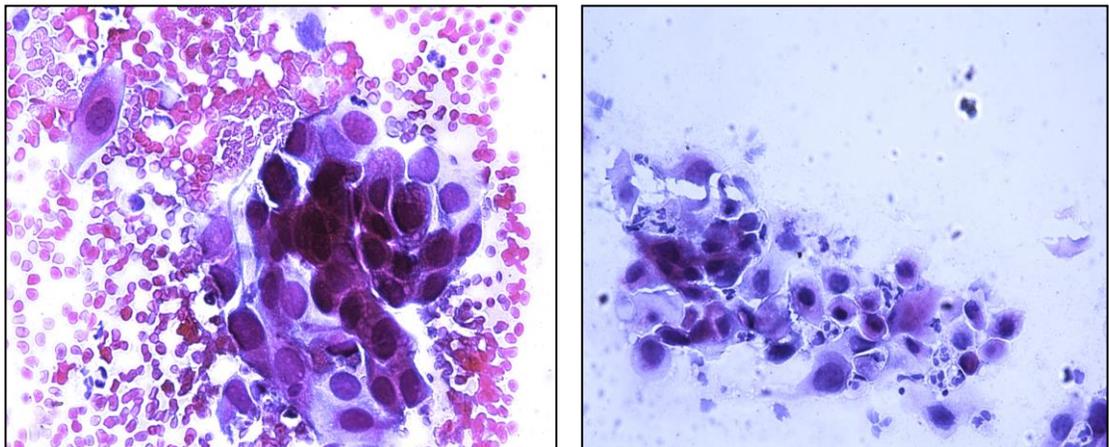


Figura 31. Representación morfológica de la clase Carcinoma (fuente: autores).

Las clases ASCUS y ASCH, suelen estar en debate entre los expertos, debido a que el diagnóstico es confundido con frecuencia para las clases contiguas, es decir LSIL y HSIL respectivamente. Lo cual establece que el cribado correcto depende la experiencia del personal.

### 3.2 Banco de Imágenes Citológicas.

La creación del banco de imágenes fue realizada en conjunto con el sector privado. Los laboratorios de la Compañía Patólogos del Cauca y Alba Luz, entregaron 50 y 30 placas citológicas respectivamente. El diagnóstico de malignidad estaba previamente determinado por patólogos especializados teniendo en cuenta el sistema estandarizado Bethesda 2001.

En este trabajo se utilizó un microscopio con cámara digital, Nikon 80 en conjunto con el software *NIS-elements* adscritos a la unidad de microscopía electrónica de la Universidad del Cauca. Los objetivos de x40 y x100 registraron las imágenes PAP con una resolución de 1600 x 1200 pixeles. Cada imagen fue clasificada por un cito histólogo en las seis clases de Bethesda. La organización del banco de imágenes para cada clase, se realizó con un orden numérico ascendente para nombrar cada una de estas (ver Figura 32).

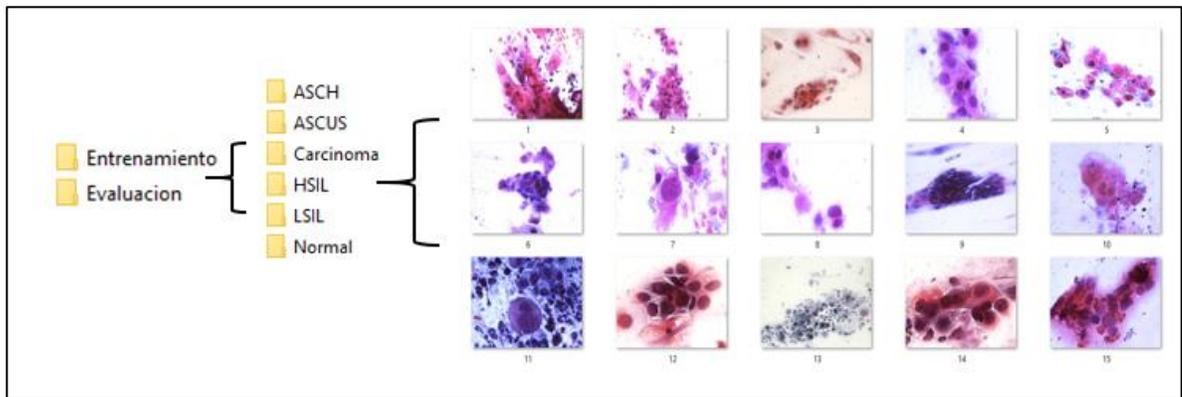


Figura 32. Estructura del banco de imágenes (fuente: autores).

Los fundamentos de la aprobación de una imagen fueron determinados por el contraste de color, claridad, área de interés, y el tipo de célula. Esta última variable hace referencia a las células escamosas de la zona exocervical para su aprobación, o a las glandulares del endocérnix, como criterio de descarte, debido a que la investigación se centra en la zona de mayor cambio celular, el exocérnix.

Para la realización de los diferentes experimentos tanto de entrenamiento como de evaluación, se utilizaron tres conjuntos, con diferentes cantidades de imágenes; el primer experimento, corresponde a una clasificación binaria con las clases normal y anormal, en donde se distribuyó aproximadamente el conjunto de imágenes con un porcentaje de entrenamiento del 71% y de evaluación del 29%. La cantidad de imágenes se muestra en la Tabla 1.

<i>Clase</i>	<i>Entrenamiento</i>	<i>Prueba</i>
	<i>Cantidad</i>	<i>Cantidad</i>
<i>Normal</i>	50	20
<i>Anormal</i>	50	20

*Tabla 1. Distribución de imágenes dos clases.*

El segundo experimento, corresponde a una clasificación multiclase desbalanceada, es decir que no posee igualdad de imágenes en las diferentes clases. En Tabla 2 se muestran las cantidades de imágenes utilizadas tanto como para la etapa de entrenamiento, como para la de evaluación y en donde se dividió la clase anormal según el sistema Bethesda para las células escamosas:

<i>Clase</i>	<i>Entrenamiento</i>	<i>Prueba</i>
	<i>Cantidad</i>	<i>Cantidad</i>
<i>Normal</i>	30	10
<i>ASCUS</i>	40	10
<i>LSIL</i>	30	10
<i>ASCH</i>	28	10
<i>HSIL</i>	29	10
<i>Carcinoma</i>	20	10

*Tabla 2. Distribución de imágenes seis clases desbalanceadas.*

El último experimento, el cual es de mayor interés, corresponde a una clasificación multiclase balanceada, es decir igual cantidad de imágenes en las diferentes clases. En la Tabla 3 se muestra la cantidad por clases:

<i>Clase</i>	<i>Entrenamiento</i>	<i>Prueba</i>
	<i>Cantidad</i>	<i>Cantidad</i>
<i>Normal</i>	30	10
<i>ASC-US</i>	30	10
<i>LSIL</i>	30	10
<i>ASCH</i>	30	10
<i>HSIL</i>	30	10
<i>Carcinoma</i>	30	10

*Tabla 3. Distribución de imágenes seis clases balanceadas.*

## 3.3 Esquema Propuesto

En este trabajo se propone representar imágenes citológicas a través de la metodología *Bag-of-Features* para clasificarlas en los grados de malignidad del estándar Bethesda 2001. Es una metodología alternativa debido a la gran complejidad que supone desarrollar un sistema de representación robusta y ligera de información visual con alto grado de similitud de las imágenes citológicas e histopatológicas en general.

Las características morfológicas entre las categorías de “células escamosas atípicas” (ASC) y las de “lesión escamosa intraepitelial” (SIL), derivan en grandes similitudes debido a que ASC-US y ASC-H son clases de una antesala para una lesión de bajo (L-SIL) y alto grado (H-SIL), respectivamente. Por lo tanto, el diagnóstico correcto de los patólogos estará enmarcado por el tiempo de experiencia y el conocimiento subjetivo. Las categorías ASC, reflejan de un modo realista la dificultad de los expertos en interpretar las muestras como lesión, además de garantizar un grado de sensibilidad y predicción. En consecuencia, se busca formas de representación visual con información altamente discriminada, de esta forma se seleccionó BoF como la metodología de implementación debido a la robustez y rapidez de caracterización. BoF, representa una imagen por la frecuencia de los parches o palabras visuales, tal como se explicó en la sección 2.4.3. No obstante, esto implica definir un conjunto de parámetros incluidos por el usuario (descrito en la sección 2.3), que permite al clasificador SVM ajustar la generalización de la información, reflejada en las métricas de clasificación. Dichos parámetros inician seleccionando el detector y descriptor, seguido por la cantidad de clústeres que determinan el tamaño del *codebook*, finalizando con la sintonización del clasificador, es decir la búsqueda de los hiper-parámetros.

El enfoque del trabajo es definido dentro del marco de la inteligencia artificial como clasificación multiclase, esto implica que está gobernado por el esquema de aprendizaje supervisado, el cual se explicó en la sección 2.2.1. En la figura 33, se expone el esquema y la interacción de las etapas divididas en dos fases: entrenamiento y evaluación. En la primera fase se consigue la construcción del *codebook* con el cual se va a comparar cada palabra visual de las imágenes de evaluación, obteniendo el histograma de representación, de igual forma, se calcula el modelo de clasificación SVM que realizará la discriminación en las clases. Finalmente, se mide el desempeño de clasificación con métricas probabilísticas, explicadas en la sección 4.1 (*Matriz de Confusión, Precisión, Recall, F1 y ROC AUC*).

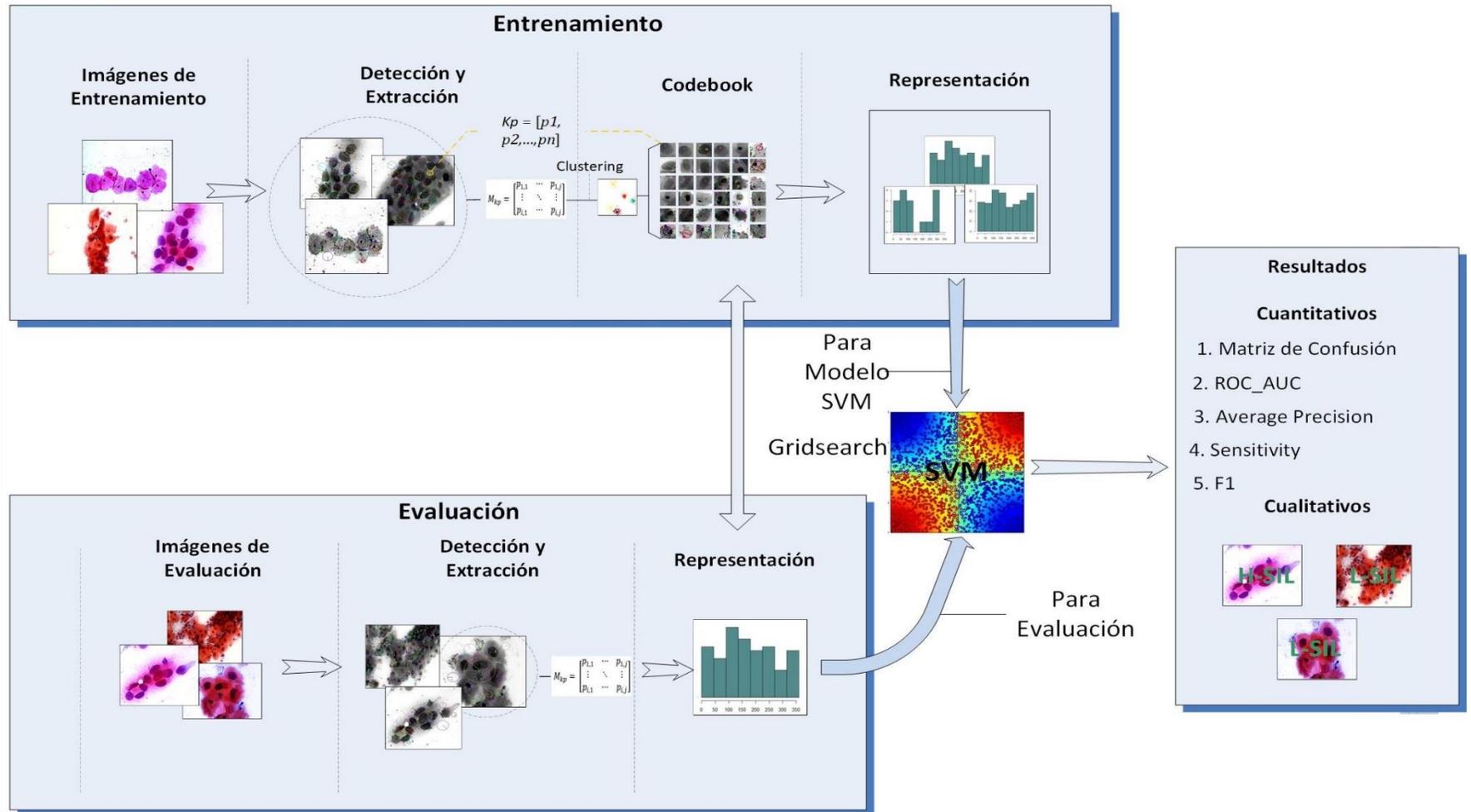


Figura 33. Esquema propuesto de Bag-of-Features (fuente: autores).

### 3.3.1 Extracción de Características

Es la primera de las tres etapas comunes del entrenamiento y evaluación, donde se detecta y extrae características relevantes. La detección es el proceso que decide donde y en que escala se muestrea una imagen, esto implica la utilización de técnicas de *operadores de puntos de interés*, tal como se explica en la sección 2.3.1.

Por otra parte, el proceso de descripción caracteriza un área alrededor de cada *keypoint*, de esta forma se generan los descriptores o vectores de características (ver Figura 34). Sin embargo, los descriptores carecen de información de color siendo esta una característica relevante, por tal razón se implementó el algoritmo *OpponentSIFT* para concatenar los datos de intensidad de los colores de las imágenes, obtenidos con el proceso de tinción. Este proceso hace referencia a *early-fusion*.

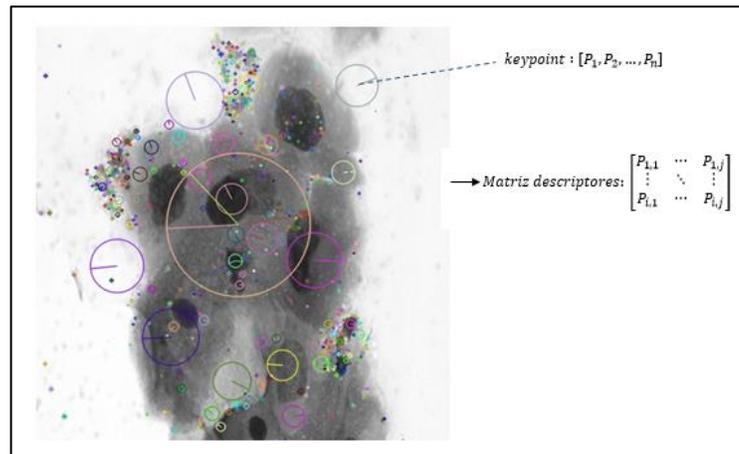


Figura 34. Resultado de la etapa de extracción (fuente: autores).

En este trabajo se evaluaron dos detectores de puntos de interés: SIFT y SURF, y cuatro descriptores: SIFT, SURF, OpponentSIFT y OpponentSURF. Se realizaron cuatro combinaciones de palabras visuales, tal como se muestra en la Tabla 4 :

<i>Extractor</i>	<i>Descriptor</i>
<i>SIFT</i>	<i>SIFT</i>
<i>SURF</i>	<i>SURF</i>
<i>SIFT</i>	<i>OpponentSIFT</i>
<i>SURF</i>	<i>OpponentSURF</i>

Tabla 4. Combinaciones Extractor/Descriptor

Debido a la aleatoriedad del número de los *keypoints* detectados por SIFT y SURF se analizaron los algoritmos para 11 tamaños del *codebook* (ver sección 3.3.2). Para este

procedimiento se tuvieron imágenes escaladas de 1600 x 1200 a 500 x 320 pixeles reduciendo la memoria de cómputo y acelerando el procesamiento.

A continuación, se muestra el pseudocódigo de la función creada en *Python* para la detección y extracción de características.

```

Función: Extracción de características
Input: imágenes entrenamiento, detector, descriptor
Output: vector de imágenes con descriptores
1:
2: for all conjunto de imágenes do
3:     leer imagen en rgb
4:     recortar imagen dimensión (500x300)
5:     convertir imagen rgb a escala de grises
6:     detector keypoints (imagen gris)
7:     if descriptor a color then
8:         calcula descriptor (imagen rgb, keypoints)
9:     else descriptor a color then
10:        calcula descriptor (imagen gris, keypoints)
11:    almacenar descriptores en un vector (imagen, descriptor)
12: end for

```

### 3.3.2 Construcción del Codebook

La cantidad total de los descriptores extraídos previamente en la fase anterior del conjunto de imágenes de entrenamiento son agrupados en  $K$  clústeres. La división se lleva a cabo mediante el cálculo de similitud entre los centroides o representantes de los clústeres y cada parche visual, el cual está determinado por una métrica de distancia euclidiana que generalmente es L1 o L2. El algoritmo más popular es *K-means*, explicado en la sección 2.4.2, y el cual se implementó en esta propuesta. Para llevar a cabo la ejecución del algoritmo es necesario realizar la sumatoria de las matrices de descriptores de cada imagen. El algoritmo de *K-means* es implementado por la librería *scikit-learn* basado en el lenguaje *Python*. Esta función tiene dos parámetros principales de entrada, matriz de *keypoints* y número de clúster o grupos. También, permite utilizar una versión avanzada del método *K-*

### 3.3 Esquema Propuesto

*means*, conocido como *K-means++* el cual se basa en la selección de los clústeres óptimos iniciales (ver Figura 35).

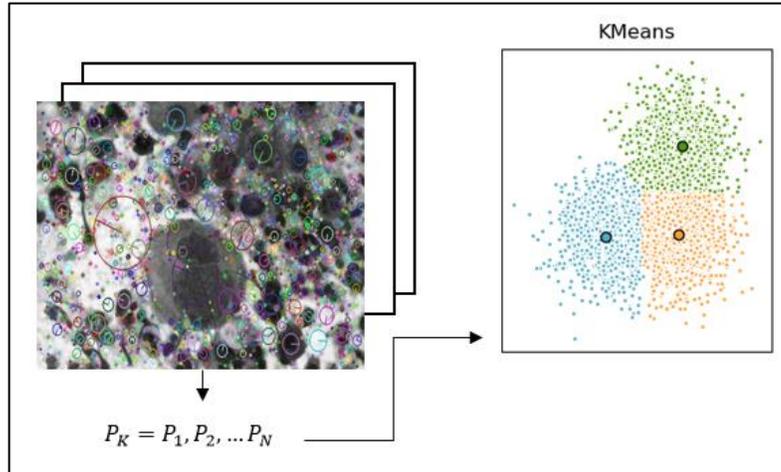


Figura 35. Agrupamiento de los descriptores por medio de *K-means++* (fuente: autores).  
A continuación, se muestra el pseudocódigo de la función para la construcción del *codebook*.

**Función:** Construcción del Codebook

**Input:** vectores de descriptores, n° de clusters

**Output:** Matriz de coordenadas de los centros de clúster

1: //

2: **for all** vectores de descriptores **do**

3:     apilar matriz de con todos los vectores

4: **end for**

5: configurar los parámetros del algoritmo Kmeans(n° de cluster)

6: aplicar algoritmo de agrupamiento Kmeans (matriz descriptores)

El aspecto más importante en esta etapa es determinar el número de clústeres que representará el tamaño del *codebook*. Un número pequeño podría limitar la cantidad variada de palabras visuales o de lo contrario sobredimensionar la representación del conjunto de imágenes de entrenamiento, todas dos situaciones indicarían un problema de generalización de información. Para determinar un número acorde al volumen de imágenes de entrenamiento

se analizaron los tamaños de 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 y 1300 para cada combinación de detector/descriptor, siendo 1300 el límite de clústeres usados debido a la alta exigencia computacional y de memoria del equipo.

### 3.3.3 Representación de Imágenes

En la última parte de la representación BoF, el número de ocurrencias de cada término dividido por el número total de las palabras visuales garantiza normalizar el conjunto de características. Esto cuantifica la importancia o influencia de cada parche visual en la imagen con valores entre 0 y 1, de esta forma la representación aproxima a una distribución de masa de probabilidad ya que la suma de frecuencias de todos los términos de la imagen es la unidad.

El método de normalización utilizado en la propuesta es la IDF frecuencia inversa de término, la cual está relacionada con la técnica TF frecuencia de términos, que en otras palabras es el conteo de las ocurrencias de las palabras visuales, así los *keypoints* con menos ocurrencias tienen mayor importancia y representan las características con mayor discriminación con respecto al resto de clases que la contienen (ver Figura 36). A continuación, se muestra el pseudocódigo de la función en *Python* para la construcción del histograma.

**Función:** Construcción del histograma.

**Input:** imágenes de entrenamiento, codebook, lista de descriptores, n° de clusters

**Output:** histograma de frecuencia de palabras visuales

```

1: //
2: for all rango de imágenes de entrenamiento do
3:     asignar el clúster correspondiente a cada descriptor
4:     for all asignación do
5:         sumatoria de ocurrencias
6:     end for
7: end for

```

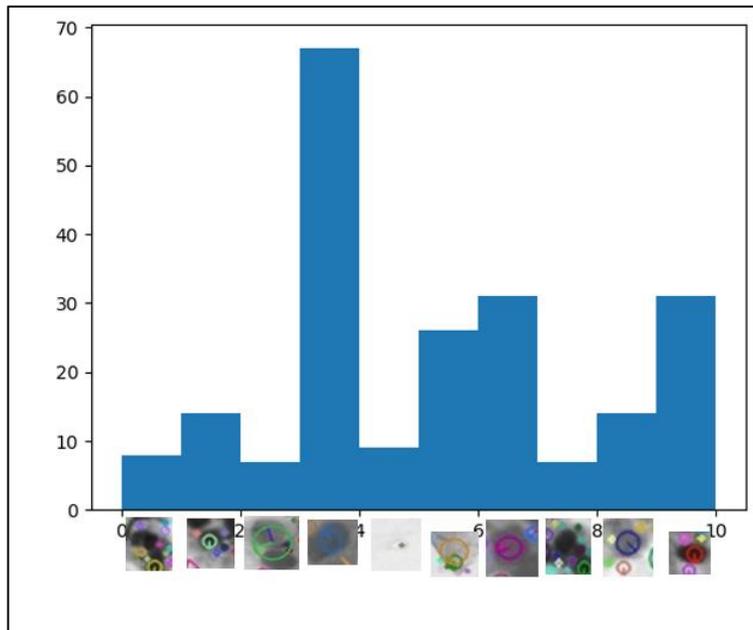


Figura 36. Ejemplo con 10 clústeres (fuente: autores).

Para la implementación de la propuesta la IDF frecuencia inversa de término se usó el siguiente comando en *Python*:

```
1: // IDF  
2: transformer: TfidfTransformer()  
3: tfidf: transformer.fit_transform(histograma)
```

En la línea 2 se hace la configuración del tipo de normalización *TF-IDF*; posteriormente en la línea 3 se ejecuta esta configuración, que logra la ponderación de las palabras más significativas y evita la dependencia del número total de palabras en la imagen.

#### 3.3.4 Clasificación de Imágenes Citológicas

Una vez se obtiene la caracterización o histograma de todas las imágenes de entrenamiento se procede con el entrenamiento del algoritmo de clasificación. Para eso, se utiliza el clasificador de SVM (*máquinas de soporte vectorial*) el cual se entrena teniendo en cuenta los siguientes pasos

- Rango de valores de hiper-parámetros.
- Método de búsqueda o muestreo de candidatos.

- Esquema de validación cruzada.
- Función de puntuación.

En este trabajo se implementó un *kernel* RBF, por lo tanto, los hiper-parámetros a calcular son  $C$  y  $\gamma$ . Con el método de *búsqueda por grilla* (*GridSearch*) en conjunto con la técnica de *Validación Cruzada* (*CV*) la cual esta implementada en *Python* gracias a la librería *scikit-learn*, se determinan los valores de los hiper-parámetros, para esto se crea un rango de valores para cada término, de esta forma se genera un conjunto de pares  $(C, \gamma)$  los cuales por cada uno se determina un puntaje o *score* con el cual está siendo evaluado (ver *Figura 25*).

El par seleccionado corresponde al mayor *score*. De esta forma, se creó un conjunto de ocho valores para los hiper-parámetros con índices crecientes de forma logarítmica de base 2, obteniendo 64 combinaciones, de igual forma, se proporcionan los índices para la validación cruzada que en este caso se utilizó de  $K$  iteraciones, lo cual permite dividir los datos en  $K$  subconjuntos de evaluación y el resto de entrenamiento. Con 10 iteraciones y los datos de entrenamiento se configuró la validación cruzada, posteriormente se ingresan el resto de los parámetros como la función de puntuación y el tipo de clasificador SVM-multiclase, *OVR*. En la Tabla 5, se resumen los valores y parámetros utilizados para el entrenamiento del clasificador SVM multiclase. Para el conjunto de valores de las variables  $C$  y  $\gamma$  se utilizó las siguientes funciones en *Python*:

```
1: // Valores C y  $\gamma$ 
2: C = np.logspace(-5,15 num=8, base=2)
3:  $\gamma$  = np.logspace(-15,3 num=8, base=2)
```

Rango C	Rango Gamma	Método de Búsqueda	Esquema CV	Función de puntuación
{ $3.125e^{-2}$ , $2.264e^{-1}$ , 1.641, $1.189e^{+1}$ , $8.614e^{+1}$ , $6.241e^{+2}$ , $4.522e^{+3}$ , $3.277e^{+4}$ }	{ $3.025e^{-5}$ , $1.814e^{-4}$ , $1.078e^{-3}$ , $6.409e^{-2}$ , $3.809e^{-2}$ , $2.264e^{-1}$ , 1.34, 8 }	<i>Grid-Search</i>	<i>K-Iteraciones</i>	<i>Recall macro</i> <i>Accuracy</i> <i>Precisión</i>

Tabla 5. Configuración de pasos para la búsqueda automático de hiper- parámetros.

### 3.3 Esquema Propuesto

A continuación, en la Tabla 6 se muestran los mejores resultados de las diferentes combinaciones detector/descriptor en las tres pruebas realizadas utilizando los clústeres detallados en la sección 3.3.2.

<i>Primer Experimento</i>					
<i>Extractor-descriptor</i>	<i>K</i>	<i>C</i>	<i>Gamma</i>	<i>Tipo SVM</i>	<i>Puntuación</i>
<i>SIFT-SIFT</i>	600	86.13	0.038	<i>OVR</i>	<i>Precisión=0.9</i>
<i>SIFT-OpponentSIFT</i>	1000	86.13	0.038	<i>OVR</i>	<i>Precisión=0.91</i>
<i>SURF-SURF</i>	900	86.13	0.038	<i>OVR</i>	<i>Precisión=0.91</i>
<i>SURF-OpponentSURF</i>	900	86.13	0.038	<i>OVR</i>	<i>Precisión=0.93</i>
<i>Segundo Experimento</i>					
<i>Extractor-descriptor</i>	<i>k</i>	<i>C</i>	<i>Gamma</i>	<i>Tipo SVM</i>	<i>Puntuación</i>
<i>SIFT-SIFT</i>	500	86.13	0.038	<i>OVR</i>	<i>Precisión=0.6</i>
<i>SIFT-OpponentSIFT</i>	700	86.13	0.226	<i>OVR</i>	<i>Precisión=0.89</i>
<i>SURF-SURF</i>	1300	32768	3.051	<i>OVR</i>	<i>Precisión=0.67</i>
<i>SURF-OpponentSURF</i>	700	11.88	0.226	<i>OVR</i>	<i>Precisión=0.64</i>
<i>Tercer Experimento</i>					
<i>Extractor-descriptor</i>	<i>k</i>	<i>C</i>	<i>Gamma</i>	<i>Tipo SVM</i>	<i>Puntuación</i>
<i>SIFT-SIFT</i>	200	11.88	1.345	<i>OVR</i>	<i>Precisión=0.57</i>
<i>SIFT-OpponentSIFT</i>	400	86.13	0.226	<i>OVR</i>	<i>Precisión=0.62</i>
<i>SURF-SURF</i>	1300	86.13	0.226	<i>OVR</i>	<i>Precisión=0.54</i>
<i>SURF-OpponentSURF</i>	700	11.88	0.038	<i>OVR</i>	<i>Precisión=0.55</i>

Tabla 6. Mejores resultados de la etapa de entrenamiento.

Una vez obtenidos los mejores modelos de entrenamiento con los hiper-parámetros, se ejecuta la predicción del clasificador SVM con las imágenes de prueba.

La evaluación del clasificador consta de ítems que son: detección y extracción de características, representación de imágenes de evaluación y por último predicción de la clasificación.

El primer y segundo ítem son realizados de igual manera que los explicados en las secciones 3.3.1 y 3.3.3 respectivamente. La construcción del *codebook* no se realiza para esta fase de evaluación; ya que es tomado de la etapa de entrenamiento con el fin de lograr agrupar los descriptores de las imágenes de prueba con los clústeres del *codebook* y así poder contabilizar la frecuencia de aparición para obtener su respectiva representación.

Una vez obtenida la representación de las imágenes de prueba, se pasa al último ítem de evaluación, en el cual se procede a realizar la comparación entre los diferentes histogramas de la etapa de entrenamiento y los de evaluación, prediciendo tanto los *labels* o *etiquetas* como los *ids* de las diferentes clases.

Los comandos utilizados para hacer predicción con SVM (*máquinas de soporte vectorial*) se describe a continuación:

```

1: // Prediccion SVM
2: predicciones_SVM = modeloEntrenamiento.predict (histograma_evaluacion)
3: resultado_predicciones = [nombre_clases[i] for i in predicciones_SVM]

```

En la línea 2 se hace la comparación de los histogramas donde se obtiene como resultado un vector de los *ids* que indican la clase a la que corresponde las diferentes imágenes de evaluación con respecto a las clases del sistema Bethesda, estos *ids* se han definido de la siguiente manera:

$$\begin{bmatrix} Asc - H \\ Asc - Us \\ Carcinoma \\ HSIL \\ LSIL \\ Normal \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

En la línea 3, se realiza la comparación de similitud, para lograr identificar si los *labels* obtenidos de las imágenes de evaluación, corresponden o pertenecen a las etiquetas de las clases reales del sistema Bethesda.

Por último, con el vector de predicciones, se procede a evaluar el desempeño del método propuesto, con diferentes métricas que logren representar y dar claridad a los resultados, los cuales son detallados en el capítulo 4.

# Capítulo 4

## Métricas y Resultados.

---

En este último capítulo se definen las métricas de validación del modelo de clasificación y los respectivos resultados para tres conjuntos diferentes de imágenes, correspondientes a dos clases, seis clases desbalanceadas y seis clases balanceadas. En la sección 4.1, se detalla las métricas más comunes para validar un sistema de clasificación las cuales se utilizan en este trabajo. En la sección 4.2, se muestra el mejor resultado por cada combinación detector/descriptor, relacionado con el número de clústeres, y la comparación de dichas combinaciones.

### 4.1. Métricas de Clasificación

La mayoría de las métricas de validación de un sistema de clasificación hacen referencia a medidas probabilísticas, las cuales tienen como objetivo indicar la similitud de un conjunto de datos, dividido en muestras de entrenamiento y evaluación. En general, las medidas dependen de la cuantificación de aciertos y errores de clasificación, calculados y mostrados en la *matriz de confusión* en unidades de porcentaje.

#### 4.1.1. Matriz de confusión

Es la métrica más común y adecuada para medir el desempeño de un sistema de clasificación supervisado. Las columnas indican los valores verdaderos presentes en los datos, mientras que las filas muestran los valores predichos por el algoritmo. El componente  $(i, j)$  establece en términos de cantidad o porcentaje, las clasificaciones correctas o incorrectas según la clase a analizar [70]. Esta información se divide en cuatro tipos de variables, tal como se muestra en la siguiente figura; **Error! No se encuentra el origen de la referencia.** que representa la estructura de la matriz de confusión (Ver Figura 37).

		Valor Predicho	
		Clase A	Clase B
Valor Verdadero	Clase A	Verdaderos Positivos	Falsos Negativos
	Clase B	Falsos Positivos	Verdaderos Negativos

Figura 37. Estructura de la matriz de confusión (fuente: autores).

- **Verdaderos Positivos (VP):** corresponde al número de aciertos para la clase positiva (Clase A).
- **Verdaderos Negativos (VN):** hace referencia al número de aciertos para la clase negativa (Clase B).
- **Falsos Negativos (FN):** son aquellas muestras clasificadas en la clase B y que en realidad pertenecen a la clase A
- **Falsos Positivos (FP):** son aquellas muestras clasificadas en la clase A y que en realidad pertenecen a la clase B.

Con base en lo anterior, una clasificación óptima es aquella con una matriz identidad, donde todos los valores de la diagonal principal correspondan a 1 y el resto a 0 [71].

#### 4.1.2. Precisión, Recall y F1

La precisión permite responder la pregunta ¿qué proporción de identificaciones positivas fue correcta para cada clase?, por lo tanto, depende de las muestras clasificadas como VP y FP.

$$Precisión = \frac{VP}{VP + FP} \quad (41)$$

Esta expresión puede tomar valores entre 0 y 1. Donde 1 indica una precisión perfecta, es decir no hay muestras clasificadas como *FP*.

La métrica *Recall*, permite indicar la proporción de muestras clasificadas correctamente, es decir *verdaderas positivas*, y al igual que la precisión su valor oscila en un rango de 0 a 1. Cuando el valor es igual 1 indica ausencia de *FN*.

$$Recall = \frac{VP}{VP + FN} \quad (42)$$

La métrica F1 es el promedio armónico de la *Precisión* y *Recall*, definida por la siguiente expresión [72].

$$F1 = 2 * \frac{precisión * recall}{precisión + recall} \quad (43)$$

### 4.1.3. Área Bajo la Curva ROC

La curva ROC es una medida visual, graficada por la tasa de verdaderos positivos (TVP) frente la tasa de falsos positivos (TFP), los cuales corresponden al eje de la ordenada y la abscisa, respectivamente, definiendo el espacio ROC (ver Figura 38 ) [73].

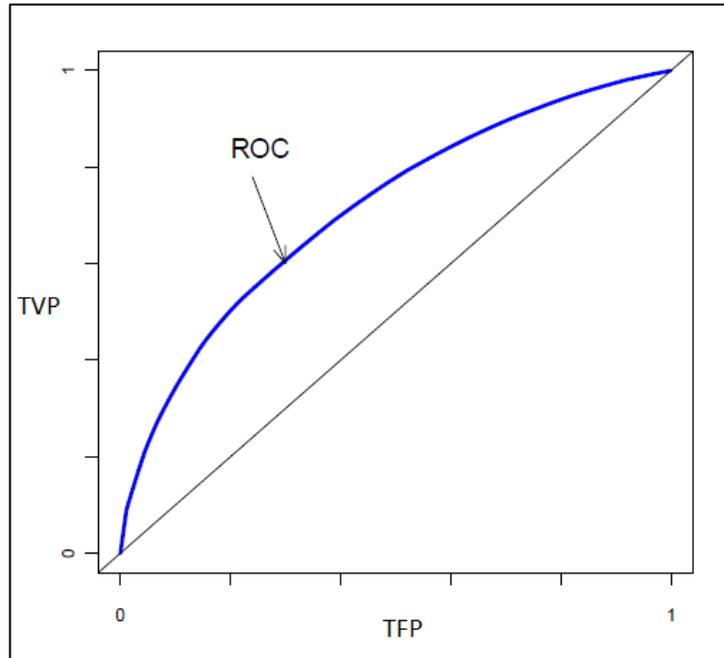


Figura 38. Ejemplo de curva ROC (fuente: adaptada de [74] ).

La tasa de verdaderos positivos es equivalente a la *sensibilidad*, la cual indica la probabilidad de una clasificación correcta si el resultado es positivo, definida como:

$$TVP = \frac{VP}{VP + FP} \quad (44)$$

Por otra parte, la tasa de falsos positivos es conocida como el complemento de la *especificidad* e indica la proporción de casos negativos que son correctamente clasificados en la evaluación, definida por la expresión 46.

$$TFP = \frac{VN}{VN + FP} \quad (45)$$

Una vez establecida la curva se calcula el área de bajo de esta, llamada comúnmente AUC (*area under curve*), la cual se interpreta como una medida de exactitud del sistema. Una exactitud alta se traduce en un desplazamiento hacia arriba y a la izquierda de la curva ROC, es decir la exactitud máxima corresponde a un valor de AUC igual a 1 y la mínima de 0.5, donde el sistema se comporta de forma indeterminada [75]. Se define como:

$$AUC = \int_0^1 ROC(d) dp \quad (46)$$

Según el criterio de Swets, indica que un área por debajo de 0.7, el sistema tiene una baja capacidad de discriminación y una por debajo de 0.9 es útil para algunas pruebas, de esta forma se establece las áreas mayores de 0.9 con alta exactitud. Sin embargo, estos criterios de interpretación son subjetivos [75], [76].

## 4.2. Experimentos y Resultados

Los experimentos corresponden al orden de los tres conjuntos de imágenes descritos en la sección **3.2** con las cuatro combinaciones de palabras visuales y la configuración de los diferentes parámetros relacionados a los algoritmos del esquema propuesto (ver Figura 33), detallados en el ítem **3.3**. La ejecución del sistema de clasificación se realizó con un PC: AMD A10, Frecuencia: 2,50GHz y RAM: 8Gb.

### 4.2.1. Experimento Uno: Dos Clases

En la Figura 39, se presenta el comportamiento de la métrica F1 versus el número de clústeres para las cuatro combinaciones.

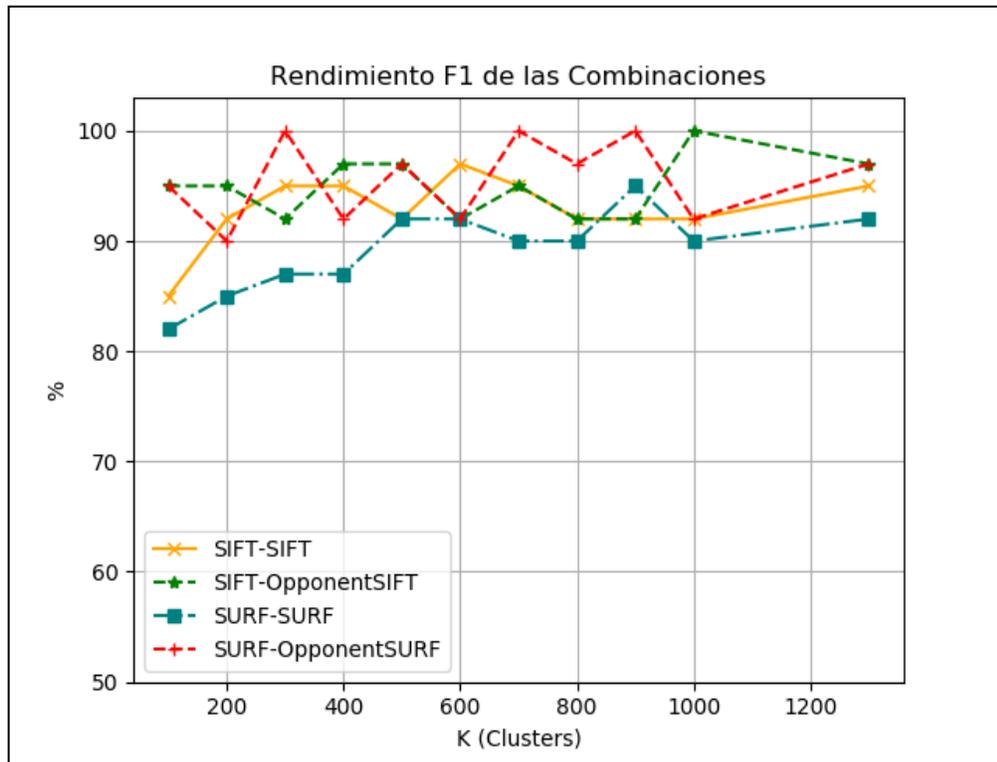
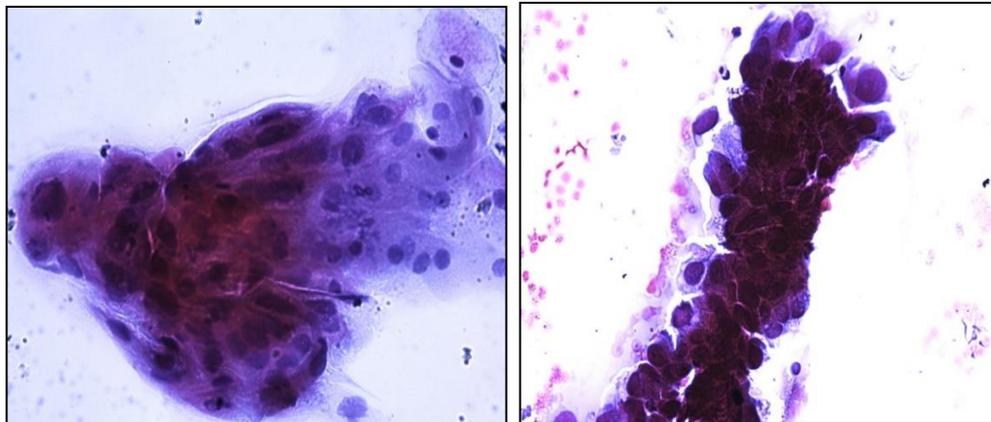


Figura 39. Comparación del rendimiento F1 para las combinaciones detector-descriptor (fuente: autores).

De la Figura 39 se aprecia una tendencia similar en la pareja de combinaciones de descripción de color, *SIFT/OpponentSIFT* y *SURF/OpponentSURF*, el cual varía entre 90% y 100% de F1, esto implica una buena precisión y *recall*, por lo tanto, la mayoría de imágenes predichas por el modelo se clasificaron en umbral de VP. Esto se debe a una buena descripción de los *keypoints* con información discriminativa de color, ya que las intensidades de tonalidad presentan gran diferencia, siendo las anormales más oscuras (ver Figura 40).



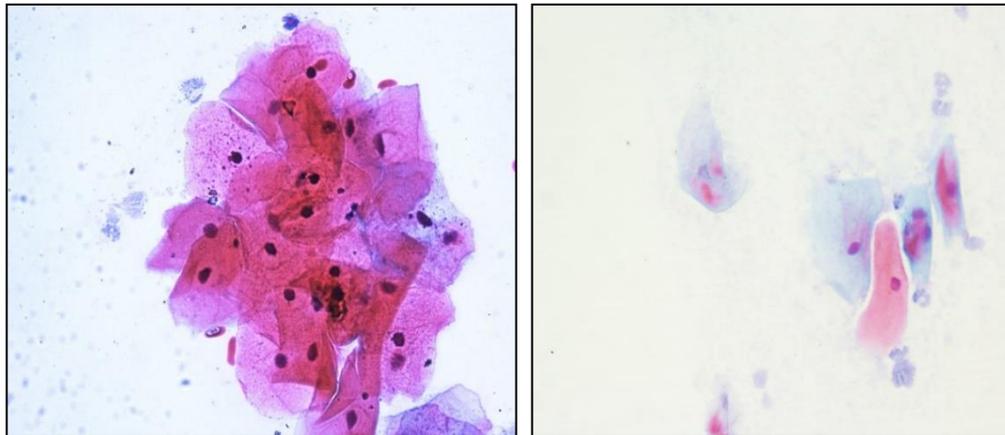


Figura 40. Imágenes utilizadas en evaluación. Las imágenes superiores son anormales y las inferiores de diagnóstico normal (fuente: autores).

Por otra parte, las dos combinaciones restantes, basadas solamente en la invariancia de escala y rotación muestran una menor utilidad de clasificación, debido a que el modelo SVM no es lo suficientemente discriminativo ante dichas características, aun así, la oscilación del porcentaje de la métrica F1 toma valores entre 80% y 95% catalogando el sistema adecuado para el diagnóstico normal o anormal, además el tiempo de procesamiento se reduce aproximadamente a la mitad del tiempo de las dos primeras combinaciones, siendo útil para grandes bancos de imágenes (ver Tabla 7).

<i>Detector / Descriptor</i>	<i>Tiempo de Entrenamiento (s)</i>
<i>SIFT / SIFT</i>	148.26
<i>SIFT / OpponentSIFT</i>	239.05
<i>SURF / SURF</i>	74.92
<i>SURF / OpponentSURF</i>	121.38

Tabla 7. Tiempo promedio de entrenamiento para cada combinación detector/descriptor.

En la Tabla 8 se resume todos los valores de Precisión (*P*), *Recall* (*R*), y *F1* (*F*), para cada instancia de *K*.

<i>K</i>	<i>SIFT / SIFT</i>			<i>SIFT / OpponentSIFT</i>			<i>SURF / SURF</i>			<i>SURF / OpponentSURF</i>		
	<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>
100	85	85	85	95	95	95	85	82	82	95	95	95
200	93	93	92	95	95	95	86	85	85	92	90	90
300	95	95	95	93	93	92	88	88	87	100	100	100
400	95	95	95	98	97	97	88	88	87	93	93	92

500	93	93	92	98	97	97	93	93	92	98	97	97
600	98	97	97	90	88	92	93	93	92	93	93	92
700	95	95	95	95	95	95	90	90	90	100	100	100
800	93	93	92	93	93	92	90	90	90	98	97	97
900	93	93	92	93	93	92	95	95	95	100	100	100
1000	93	93	92	100	100	100	92	90	90	93	93	92
1300	95	95	95	98	97	97	93	93	92	98	97	97

Tabla 8. Métricas de rendimiento, precisión, recall y F1 para clasificación binaria.

Los valores marcados con color rojo indican los mejores resultados para cada detector/descriptor. Promediando las métricas se escoge la mejor combinación (ver Tabla 9), la cual se estudiará con mayor detalle de aquí en adelante.

Combinación	P	R	F
SIFT SIFT	93.54	93.36	92.9
SIFT OpponentSIFT	95.27	94.81	94.9
SURF SURF	90.27	89.72	89.27
SURF OpponentSURF	96.36	95.9	95.63

Tabla 9. Promedio de las métricas para cada combinación.

De la Tabla 9, se observa que SURF/OpponentSURF tiene una gran ventaja en clasificar imágenes normales y anormales, El comportamiento de las métricas se muestra en las siguientes tres figuras (Figura 41, Figura 42 y Figura 43). Por otra parte, SIFT/OpponentSIFT se posiciona en segundo lugar con el mejor rendimiento.

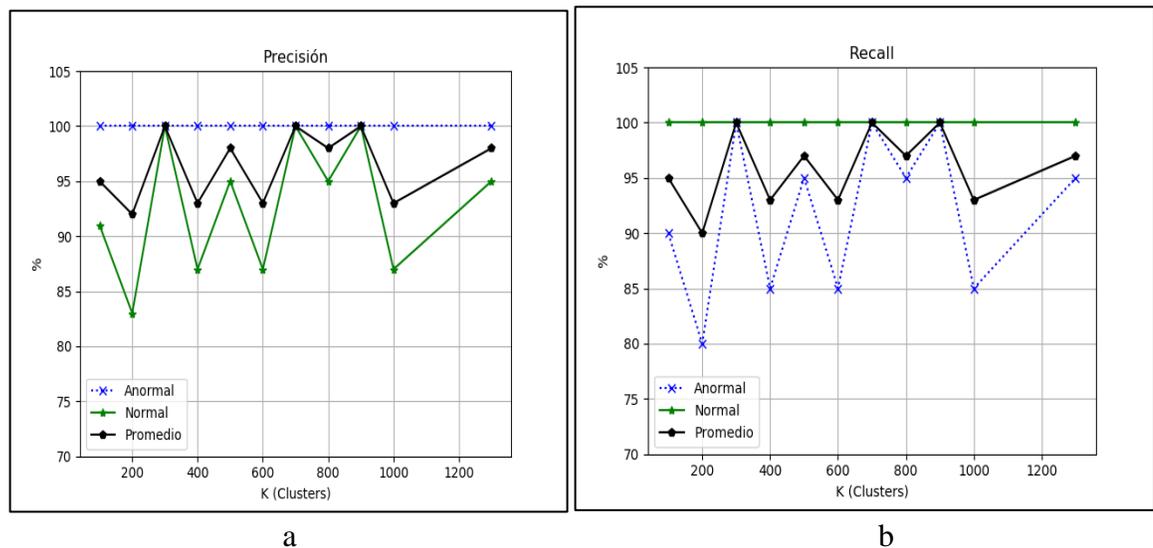


Figura 41. **a.** Comportamiento de la métrica precisión, **b.** comportamiento de la métrica recall, en un tipo detector/descriptor, SURF/OpponentSURF, variando el valor de  $K$  (fuente: autores).

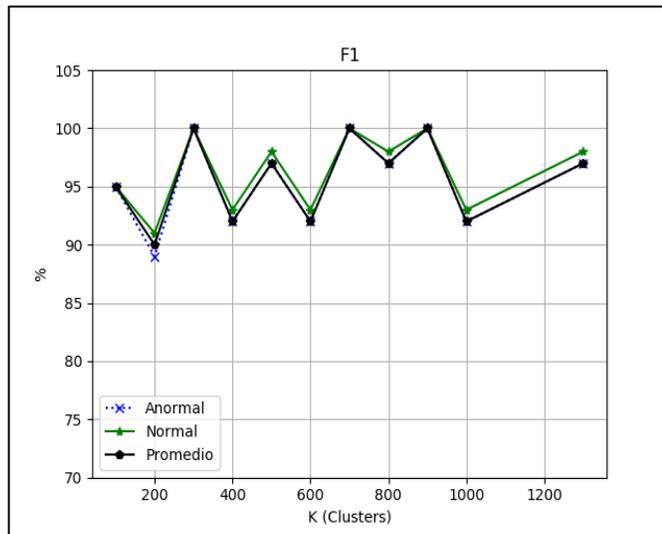


Figura 42. Comportamiento de la métrica F1 en un tipo detector/descriptor, SURF/OpponentSURF, variando el valor de K (fuente: autores).

De la traza de la Figura 42 se expone un comportamiento similar para las dos clases, con un alto grado de discriminación, esto significa que una buena aproximación de la cantidad de imágenes son correctamente separadas por cada clase. En el caso particular un tamaño de *codebook* igual a 300, 700 y 900 clústeres se evidencia una clasificación perfecta, tal como lo resume la Figura 43 con la matriz de confusión para el último caso.

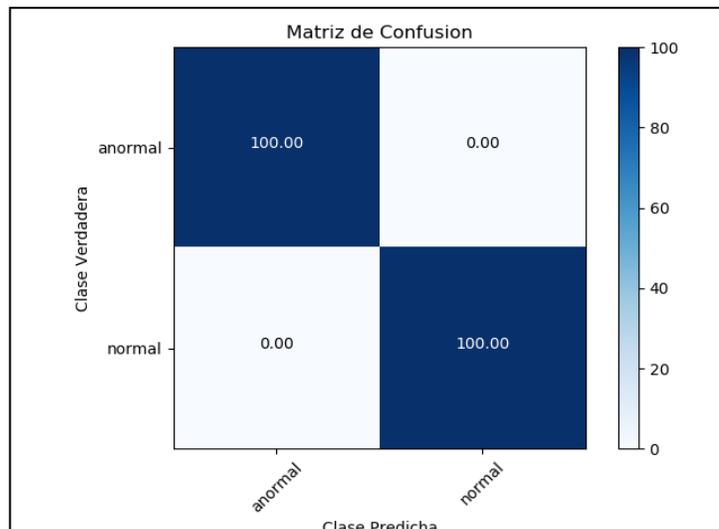


Figura 43. Matriz de confusión para una combinación SURF/OpponentSURF con un tamaño de *codebook* igual a 900 (fuente: autores).

En la Figura 44, se resume el comportamiento de la métrica ROC AUC en todas las instancias de K para la mejor combinación. Por último, se muestra la curva ROC con su correspondiente

valor AUC de cada clase y el promedio indicando el rendimiento del modelo de clasificación, para un tamaño de *codebook* igual a 900 clústeres, correspondiente al mejor resultado.

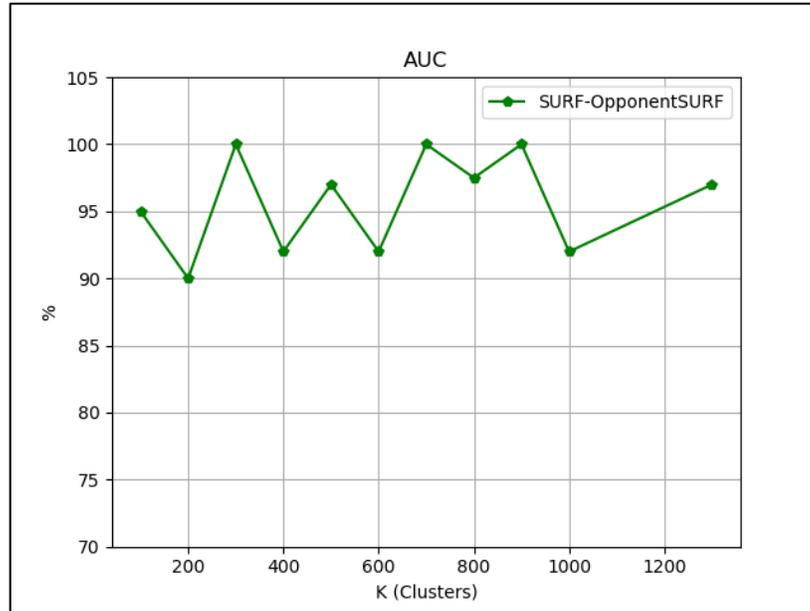


Figura 44. Comportamiento de la métrica ROC AUC de SURF/OpponentSURF, para cada  $K$  (fuente autores).

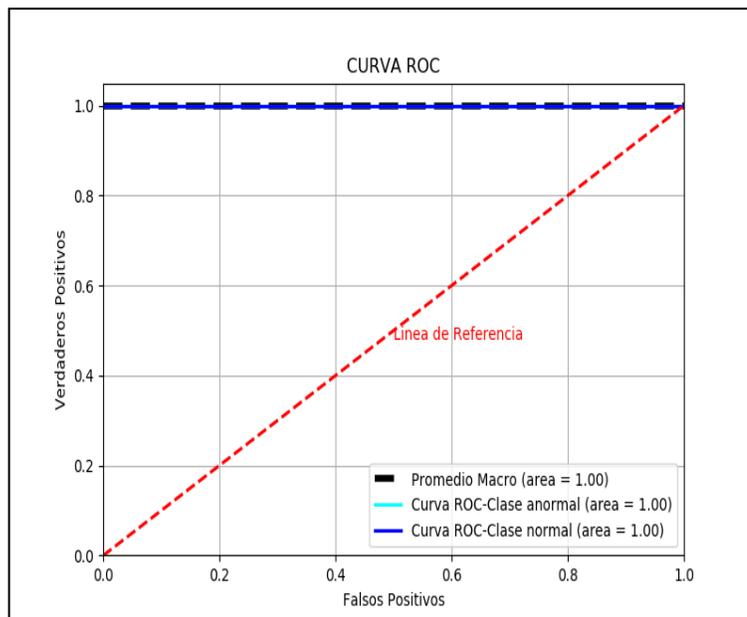


Figura 45. Curva ROC AUC de SURF/OpponentSURF con  $K$  igual a 900 (fuente: autores).

El índice AUC representa la probabilidad de que una imagen aleatoria normal o anormal sea clasificada en la clase correspondiente. En la Figura 45 se muestra que se logra el mejor valor de AUC igual a 100% para cada clase, es decir el modelo SVM clasificó todas las imágenes en la clase correcta, además presenta un promedio como índice de rendimiento del sistema. En este caso el sistema es determinado como óptimo.

### 4.2.2. Experimento Dos: Seis Clases Desbalanceadas

El comportamiento del rendimiento del clasificador con los cuatro tipos de palabras visuales a través de las combinaciones, se resume en la Figura 46 con la tendencia de la métrica F1.

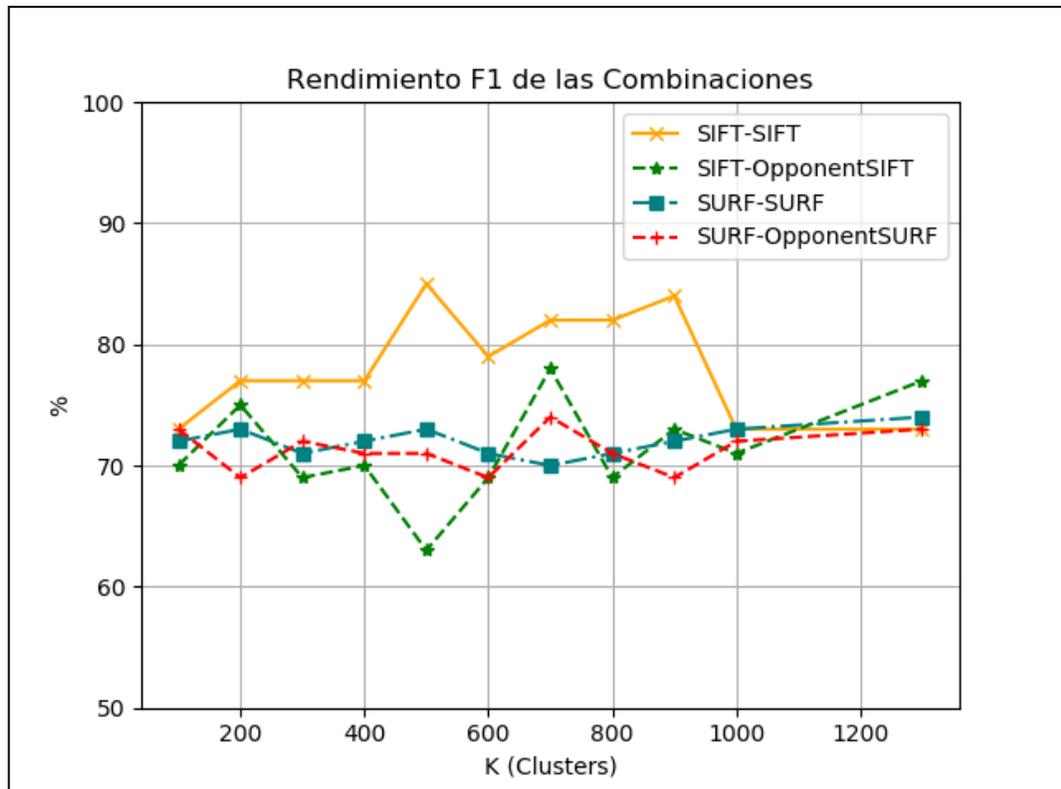


Figura 46. Comportamiento de las diferentes combinaciones con la métrica F1 (fuente: autores).

De la figura anterior se evidencia una disminución en el rendimiento del modelo de clasificación en todas las combinaciones comparándola con el experimento uno, esto se debe al aumento de clases, siendo este un resultado esperado. Sin embargo, la combinación *SIFT/SIFT* se destaca en las instancias de *K* desde 200 a 900, lo cual implica una clasificación con mejor precisión y *recall*, en otras palabras, menor cantidad de imágenes categorizada en

los umbrales de FP y FN. Por otra parte, el resto de *detectores/descriptores* describen un comportamiento estable del modelo, los cuales fluctúan entre 69% y 75% aproximadamente. Las combinaciones que representan las intensidades de color tuvieron una menor precisión y *recall*, debido a que las muestras de las clases anormales presentan información muy similar, tal y como se aprecia en algunas de las siguientes imágenes de evaluación (ver Figura 47).

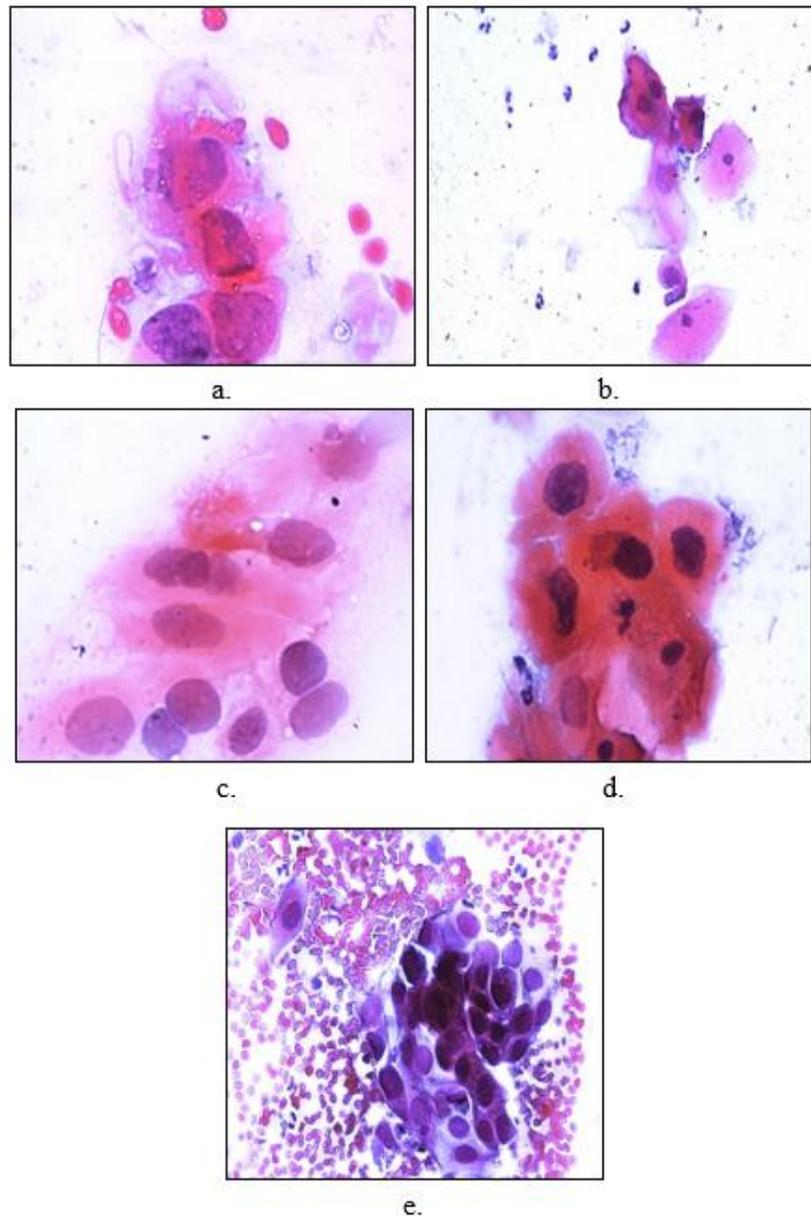


Figura 47. Imágenes con células anormales. *a.* ASCH, *b.* ASCUS, *c.* HSIL, *d.* LSIL, *e.* Carcinoma (fuente: autores).

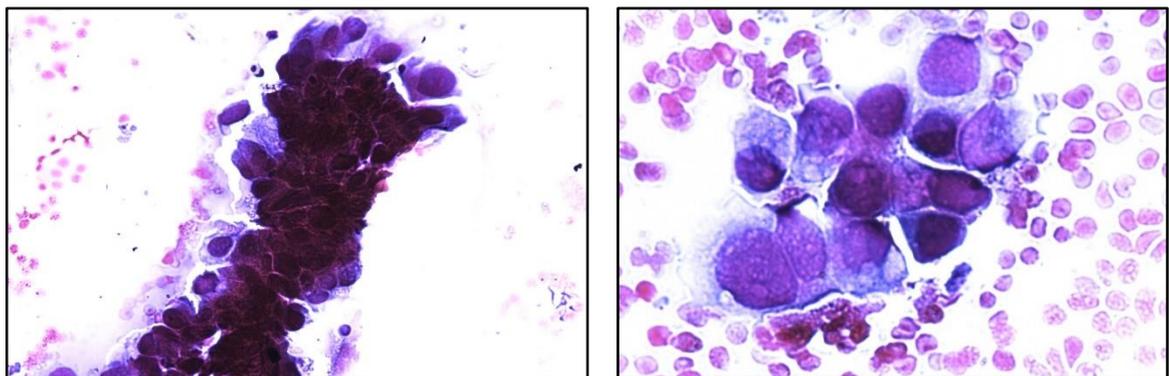
## 4.2. Experimentos y Resultados

En la Tabla 10 se resume todos los valores de Precisión ( $P$ ),  $Recall$  ( $R$ ), y  $F1$  ( $F$ ), para cada instancia de  $K$ .

$K$	<i>SIFT</i> <i>SIFT</i>			<i>SIFT</i> <i>OpponentSIFT</i>			<i>SURF</i> <i>SURF</i>			<i>SURF</i> <i>OpponentSURF</i>		
	$P$	$R$	$F$	$P$	$R$	$F$	$P$	$R$	$F$	$P$	$R$	$F$
100	78	72	73	71	72	70	74	72	72	75	73	73
200	79	77	77	77	75	75	74	73	73	69	70	69
300	80	77	77	72	70	69	70	72	71	75	73	72
400	78	77	77	74	70	70	73	73	72	71	72	71
500	85	85	85	64	63	63	74	73	73	72	72	71
600	80	78	79	71	70	69	72	72	71	70	70	69
700	83	82	82	80	78	78	70	70	70	76	75	74
800	83	82	82	71	70	69	71	72	71	73	72	71
900	84	83	84	74	73	73	73	73	72	73	70	69
1000	74	73	73	74	72	71	73	73	73	74	73	72
1300	75	73	73	77	77	77	75	75	74	73	72	73

Tabla 10. Métricas de rendimiento, precisión, recall y  $F1$  de clasificación multiclase.

La representación de *SIFT/SIFT* con 500 clústeres se destaca por alcanzar la mejor clasificación de este experimento, mientras que para *SURF/SURF* se necesitaron la mayoría de clústeres para lograr los resultados más altos. Por otra parte, con un  $K$  igual a 700 se consiguió las mejores métricas para los descriptores de color, sin embargo, estas dos combinaciones no superaron el rendimiento de la primera combinación, ya que algunas imágenes de la misma clase no cumplían con el tipo de tinción adecuado, derivando en tonalidades más intensas que otras, además, la superposición celular presenta mayor densidad de color, tal como se muestra el comparativo de la Figura 48. Finalmente, la Tabla 11 indica el resumen del rendimiento del modelo de clasificación para cada combinación.



a.

b.

Figura 48. a. Carcinoma con mayor densidad de células superpobladas b. Carcinoma con células dispersas (fuente: autores).

Combinación	<i>P</i>	<i>R</i>	<i>F</i>
<i>SIFT/SIFT</i>	79.9	78.7	78.36
<i>SIFT/OpponentSIFT</i>	73.18	71.81	71.27
<i>SURF/SURF</i>	72.63	72.54	72
<i>SURF/OpponentSURF</i>	72.18	72	71.27

Tabla 11. Promedio de las métricas de clasificación en cada combinación.

Enfatizando en la combinación *SIFT/SIFT* se presenta la traza de comportamiento de F1 en la Figura 49.

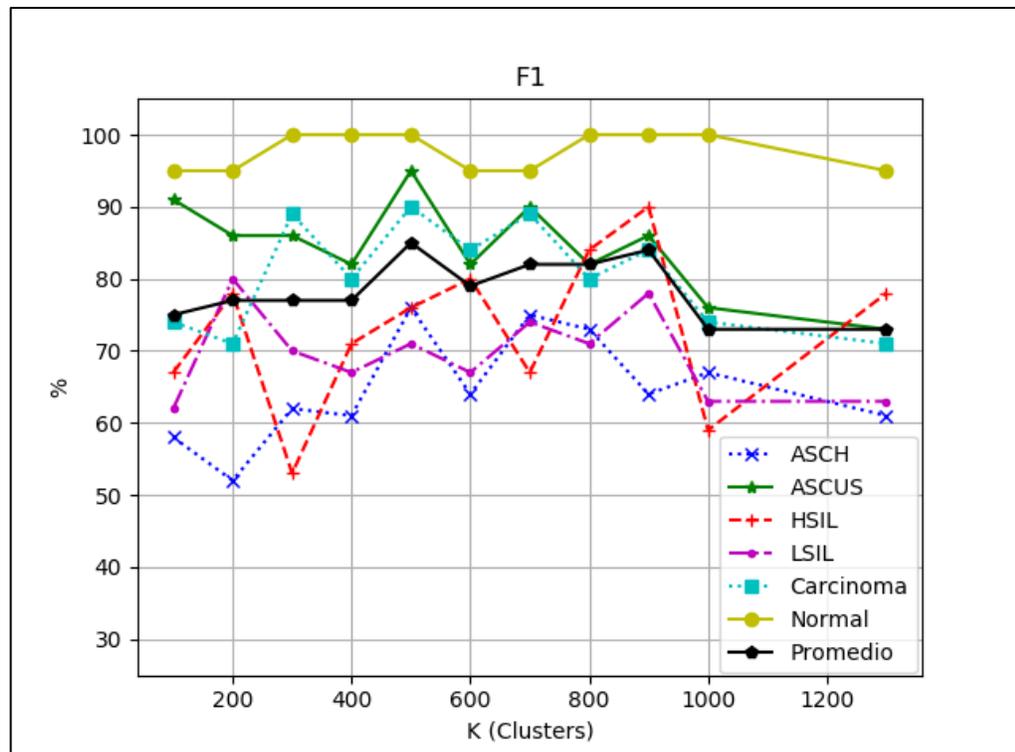


Figura 49. Comportamiento de la métrica F1 en las clases, variando el valor de *K* para la combinación *SIFT/SIFT* (fuente: autores).

Teniendo en cuenta la tendencia del promedio (línea negra), la variación del tamaño del *codebook* no representa una variable de mayor incidencia para la clasificación, ya que cuatro de las clases (ASCUS, HSIL, Carcinoma y Normal) se consideran de comportamiento estable, caso contrario a LSIL y ASCH donde se evidencia altibajos pronunciados.

## 4.2. Experimentos y Resultados

La probabilidad de evaluar una imagen de cualquier clase y que esta sea clasificada correctamente se resume con la métrica ROC AUC. En la Figura 50 donde se presenta el comportamiento del promedio del área bajo la curva ROC para cada tamaño de *codebook*.

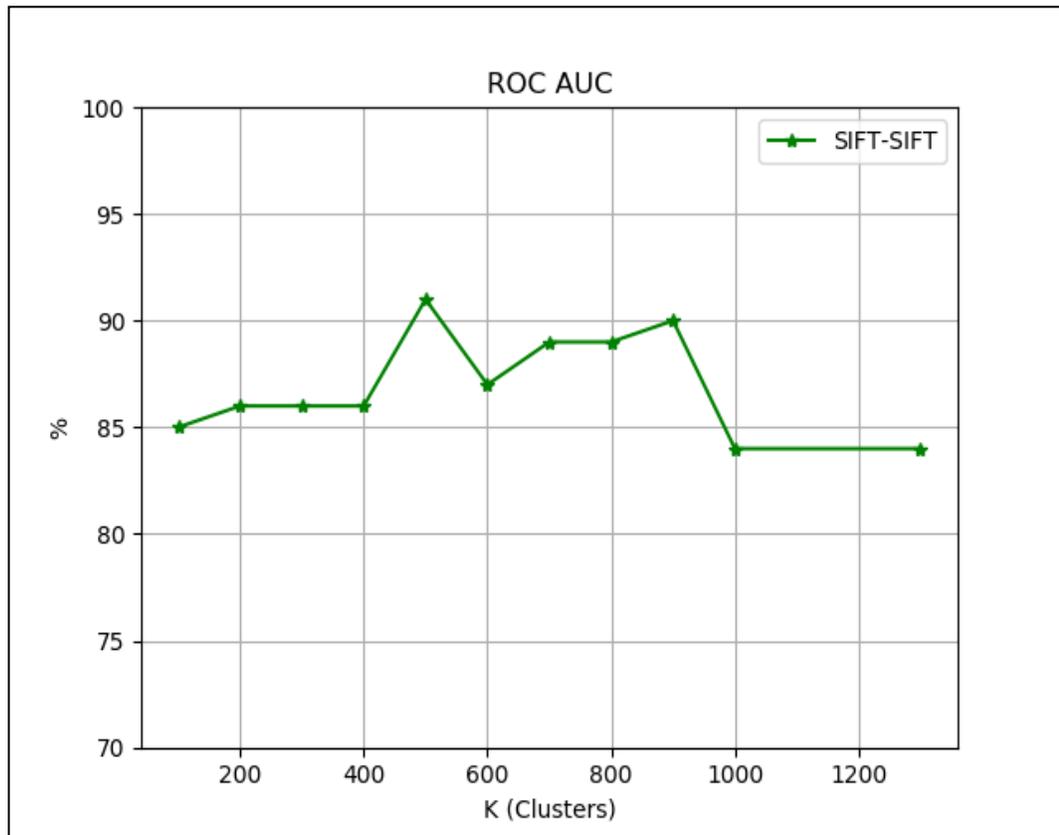


Figura 50. Comportamiento de la métrica ROC AUC en SIFT/SIFT, para cada K (fuente: autores).

La diferencia porcentual de los 8 valores de K, a partir de 200, con respecto a un índice de AUC igual a 85% como referencia, es de un dígito, caracterizando el modelo como estable, sin embargo, este tipo de traza sesga parte de información de las clases, por tal razón se expone la curva ROC para un tamaño de *codebook* igual 500 clústeres, siendo el rendimiento más alto (ver Figura 51).

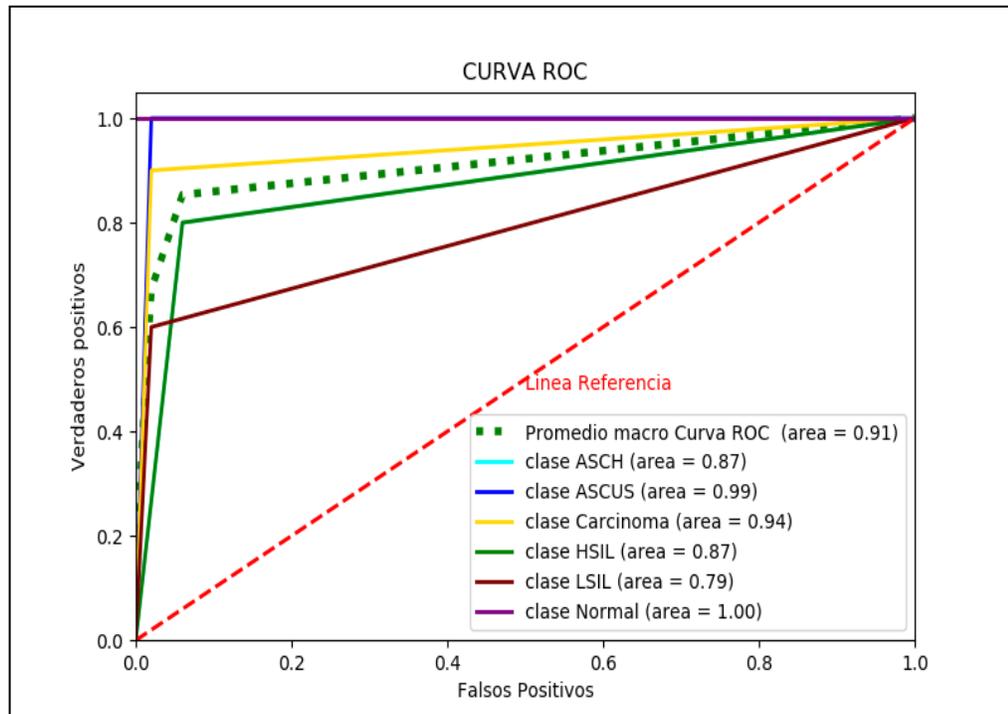


Figura 51. Curva ROC AUC con un  $K$  igual a 500 y una combinación SIFT/SIFT (fuente: autores).

El modelo de clasificación obtuvo un AUC igual a 91%, garantizando una clasificación con gran precisión y *recall*. Esto se debe al hecho que la información de entrenamiento crea un modelo SVM lo suficientemente discriminativo para separar 5 clases con un índice AUC superior al 85%. La principal conclusión de esta experimentación se basa en el conocimiento principal que suele residir en las clases menos representadas [77]. Tal y como lo evidencia la matriz de confusión (ver Figura 52).

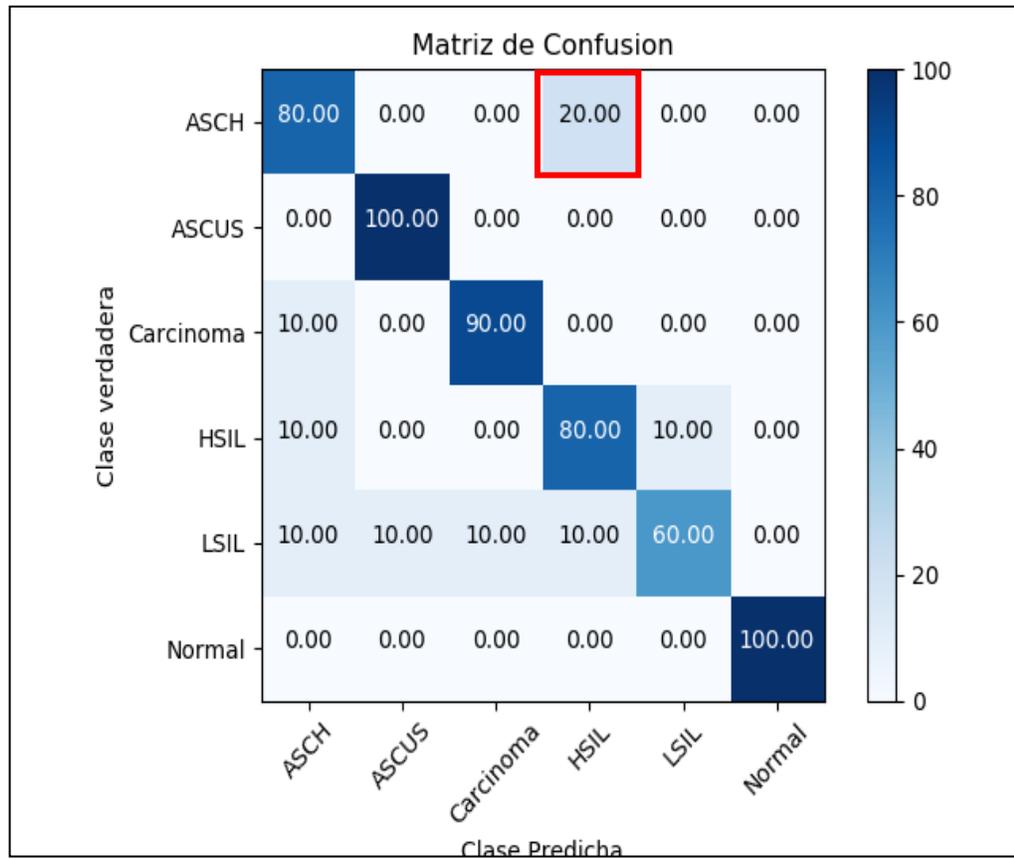


Figura 52. Matriz de confusión de seis clases desbalanceadas (fuente autores).

A pesar de tener clases con más número de imágenes de entrenamiento (ver Tabla 2 ) el modelo debe ser capaz de clasificar la mayor cantidad imágenes correctamente. Este resultado se observa en la clase Carcinoma, con 20 imágenes entrenamiento obtuvo solo 1 imagen falsa, así mismo, se resalta las clases Normal y ASCUS con la totalidad de las imágenes en los umbrales de clasificación correcta. En la clase ASCH se presenta 20% de imágenes falsas (cuadro de color rojo), sin embargo, estas entran en el umbral de error de decisión de los expertos, debido a que HSIL comparte características similares. Por último, la clase LSIL obtuvo la menor cantidad de aciertos, con un total de 6.

### 4.2.3. Experimento Tres: Seis Clases Balanceadas

En este último experimento se equilibran la cantidad de imágenes con el objetivo de evitar prioridades de discriminación, de esta forma se muestra la Figura 53 donde se traza el comportamiento de la métrica F1 para la descripción de los cuatro tipos de palabras.

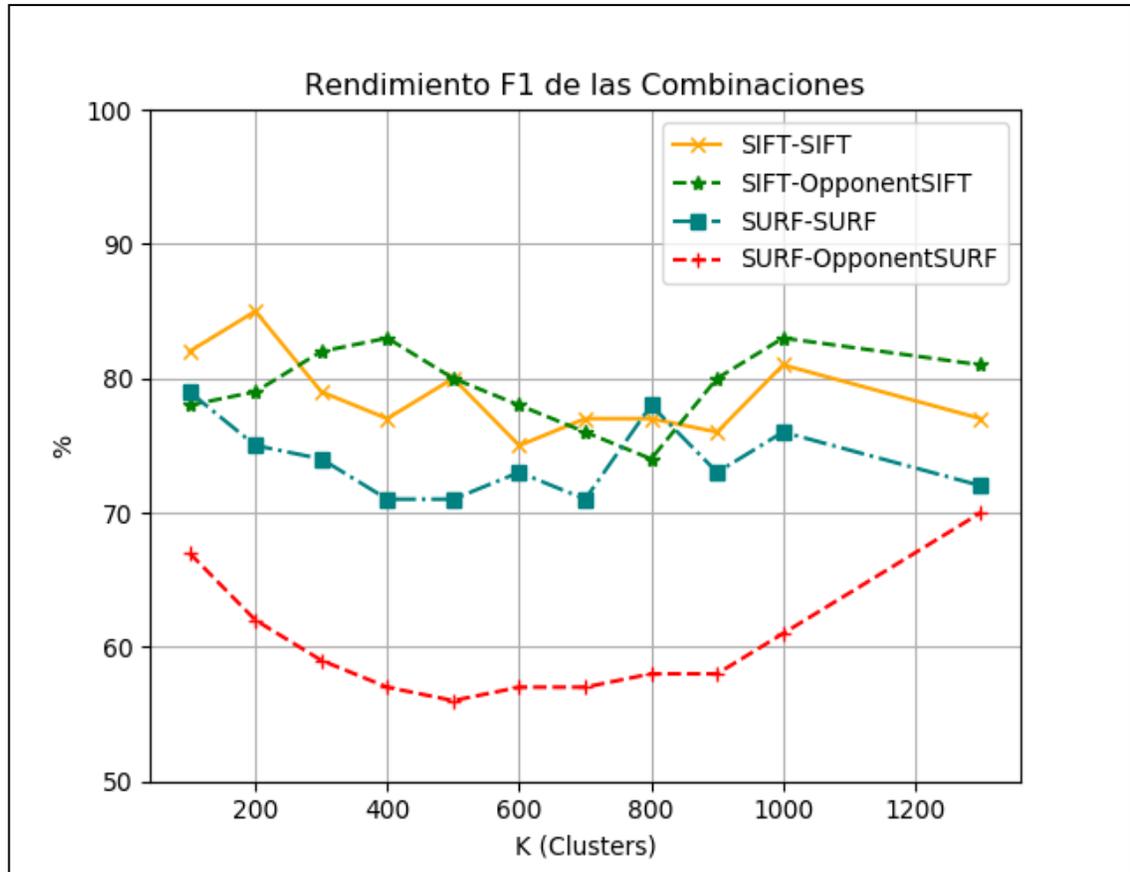


Figura 53. Comportamiento de las combinaciones con la métrica F1 (fuente: autores).

La Figura 53 indica una notable disminución en la clasificación de las imágenes para la combinación *SURF/OpponentSURF*, donde el comportamiento grafica una especie de “U”, siendo el 70% el valor más alto, correspondiente a un tamaño de *codebook* igual a 1300 clústeres. Por el contrario, las otras tres combinaciones tienen un comportamiento similar, por esta razón se detalla los resultados de las métricas en la Tabla 12 y el promedio de estos en la Tabla 13.

## 4.2. Experimentos y Resultados

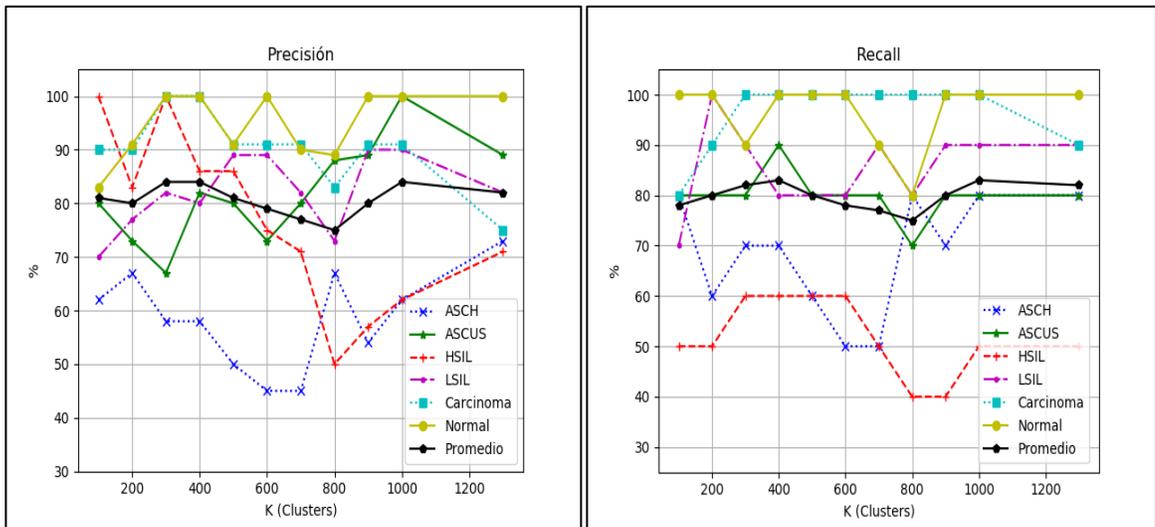
<i>K</i>	<i>SIFT</i> <i>SIFT</i>			<i>SIFT</i> <i>OpponentSIFT</i>			<i>SURF</i> <i>SURF</i>			<i>SURF</i> <i>OpponentSURF</i>		
	<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>	<i>P</i>	<i>R</i>	<i>F</i>
100	83	82	82	81	78	78	84	80	79	73	68	67
200	86	85	85	80	80	79	75	75	75	68	63	62
300	80	78	79	84	82	82	75	75	74	69	60	59
400	80	77	77	84	83	83	72	70	71	73	58	57
500	81	80	80	81	80	80	73	72	71	75	57	56
600	76	75	75	79	78	78	78	73	73	72	57	57
700	78	77	77	77	77	76	72	72	71	75	58	57
800	80	77	77	75	75	74	80	78	78	76	58	58
900	77	77	76	80	80	80	73	73	73	71	57	58
1000	82	82	81	84	83	83	77	77	76	73	60	61
1300	77	77	77	82	82	81	72	72	72	75	72	70

Tabla 12. Métricas de rendimiento, precisión, recall y F1 de clasificación multiclase.

<i>Combinación</i>	<i>P</i>	<i>R</i>	<i>F</i>
<i>SIFT SIFT</i>	80	78.81	78.72
<i>SIFT OpponentSIFT</i>	80.63	79.81	79.45
<i>SURF SURF</i>	75.77	74.22	73.88
<i>SURF OpponentSURF</i>	72.5	59.6	59.2

Tabla 13. Promedio de las métricas de clasificación en cada combinación.

Teniendo en cuenta las anteriores tablas se observa que la combinación con mejor desempeño es *SIFT/OpponentSIFT*, con una instancia de *K* igual a 400 y 1000 se obtuvo las métricas más altas. Sin embargo, la clasificación con las mayores métricas a nivel general se encuentra en *SIFT/SIFT* con un tamaño de *codebook* igual a 200, dado que los operadores de puntos de interés detectan de forma aleatoria, por lo tanto, la cantidad de variaciones de *keypoints* se aproxima considerablemente al tamaño del *codebook* obteniendo una representación buena. Por otra parte, se tiene el bajo desempeño de la última combinación con valores de métricas inferiores a los promedios del experimento dos. A continuación, se visualiza el comportamiento de las tres métricas para la combinación más destacada (ver Figura 54, Figura 55).



a.

b.

Figura 54. **a.** Comportamiento de la métrica precisión **b.** comportamiento de la métrica recall en un tipo detector/descriptor, SIFT/OpponentSIFT, variando el valor de K (fuente: autores).

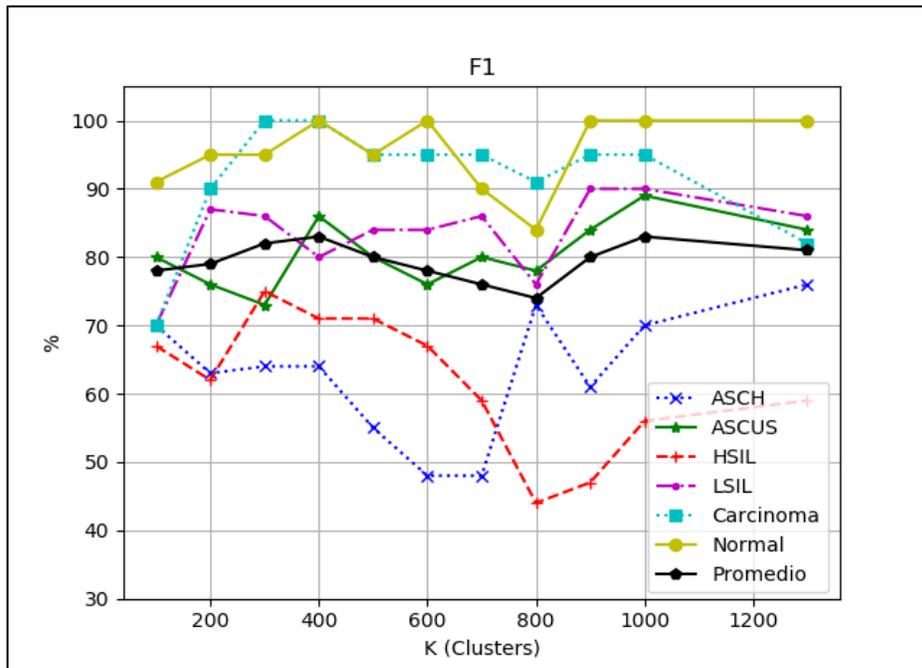


Figura 55. Comportamiento de la métrica F1 en un tipo detector/descriptor, SIFT/OpponentSIFT, variando el valor de K (fuente: autores).

Las clases de los extremos en grado de malignidad celular, normal y carcinoma, muestran mayor valor de F1, es decir, se presentan menor cantidad de imágenes predichas en umbrales de FP y FN. Mientras que LSIL y ASCUS tienen comportamiento similar, oscilando en

valores desde 74% hasta 90%. Por último, en las clases con un diagnóstico más severo, ASCH y HSIL, se obtuvo una menor precisión y *recall*.

En la Figura 56 se muestra el comportamiento del promedio de los valores AUC en cada tamaño del *codebook*. Verificando la clasificación más alta en 400 y 1000 clústeres.

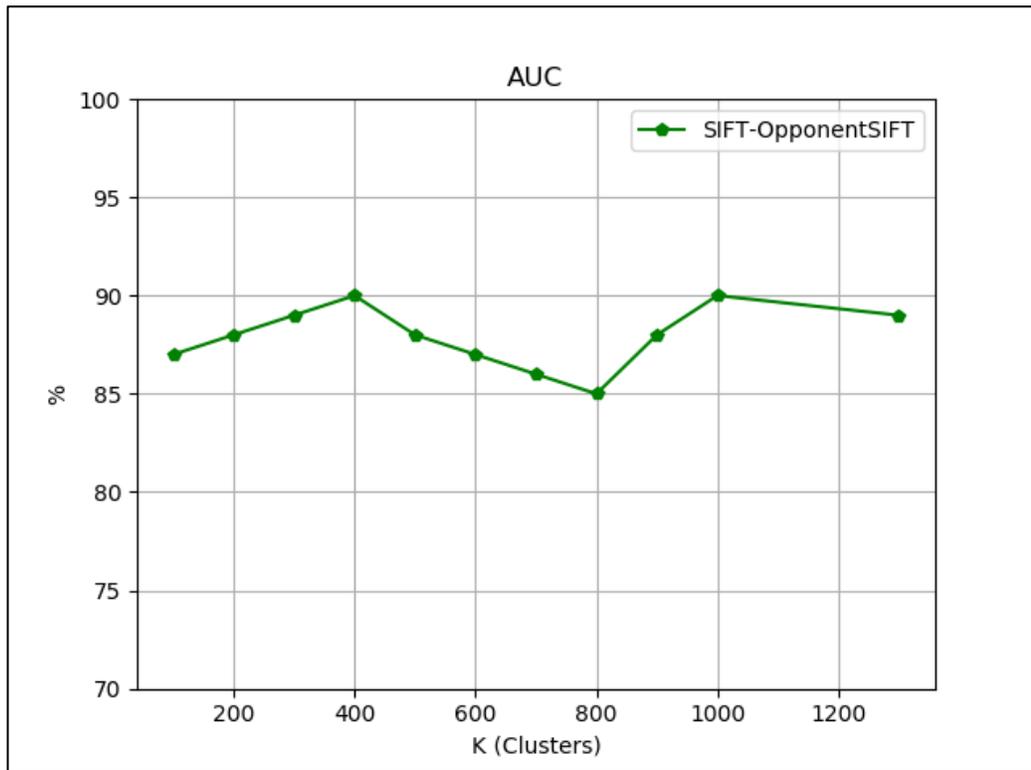
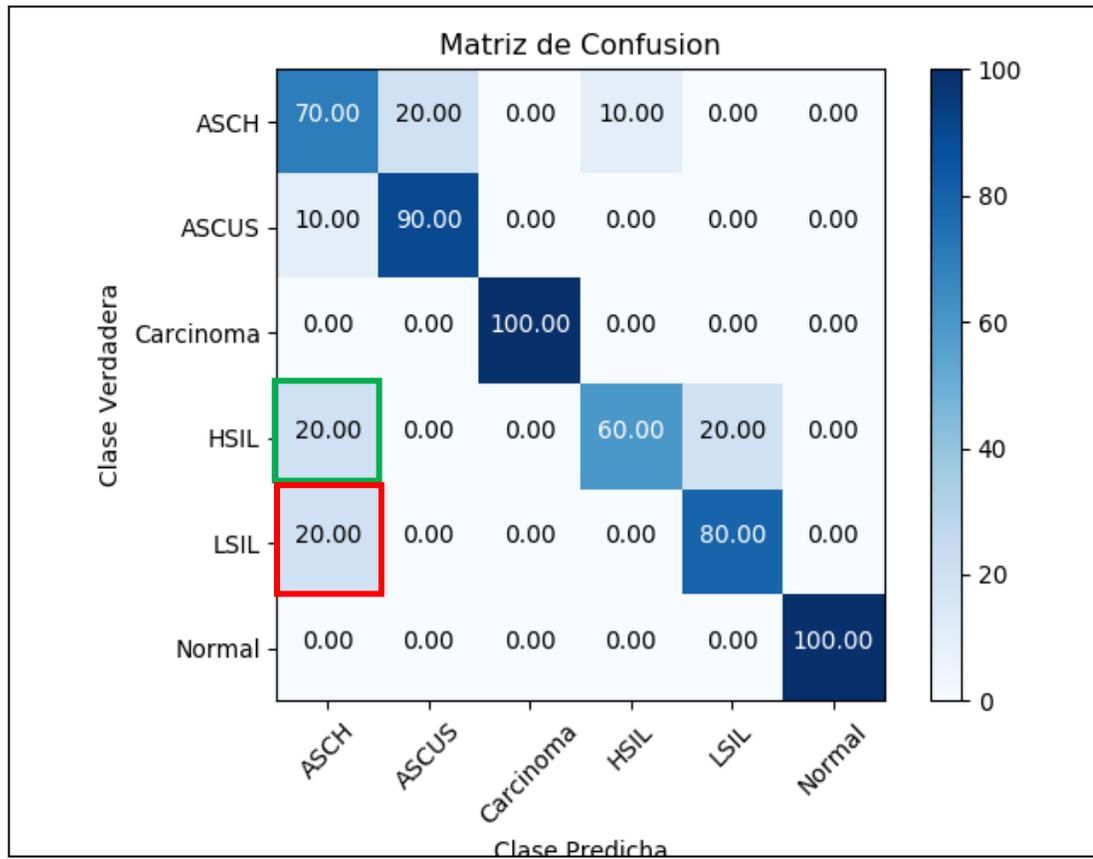


Figura 56. Comportamiento de la métrica AUC en, SIFT/ OpponentSIFT, para cada K (fuente: autores).

Promediando los anteriores valores de AUC se obtiene un valor de 87.88% deduciendo una buena caracterización de las imágenes con SIFT/OpponentSIFT para conseguir un modelo de clasificación lo suficientemente discriminativo. En la Figura 57 se muestra la matriz de confusión para el caso particular correspondiente a un tamaño de *codebook* de 400 clústeres.



*Figura 57. Matriz de confusión de seis clases balanceadas con una combinación SIFT/OpponentSIFT y un tamaño de codebook a 400 (fuente: autores).*

Tal y como lo indica la Figura 55 y se comprueba en la Figura 57, las clases normal y carcinoma, se obtuvo la totalidad de las imágenes de evaluación en el umbral de VP. Para la clase HSIL donde se predijó dos imágenes en la clase ASCH (cuadro de color verde) esto se considera un diagnóstico en el umbral de error por el experto, debido a que son clases contiguas, caso contrario a las dos imágenes predichas en ASCH (cuadro de color rojo) que realmente pertenecían a LSIL, donde las características morfológicas tienen mayor diferenciación.

Finalmente, se presenta el rendimiento del modelo de clasificación con la métrica ROC AUC para el mismo caso, en la Figura 58.

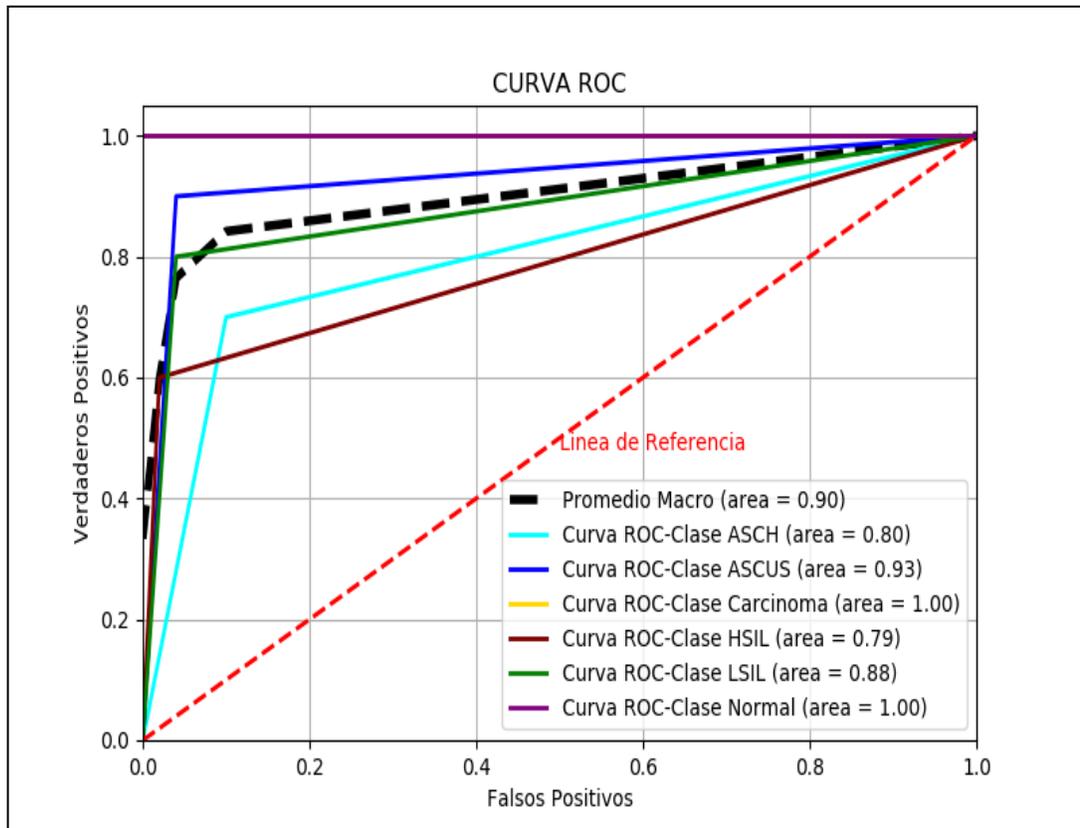


Figura 58. Curva ROC AUC con un  $K=400$  y una combinación SIFT/OpponentSIFT (fuente: autores).

Teniendo en cuenta los criterios Sweet se establece que las clases, Carcinoma, Normal y ASCUS, tiene una buena exactitud, lo cual representa una alta probabilidad de que el sistema clasifique las imágenes en las clases correctamente. Para las otras tres clases restantes se obtuvo un área considerable para discriminar la mayoría de las imágenes de evaluación en las clases correspondientes, sin embargo, HSIL presenta la menor probabilidad de que una imagen al azar sea clasificada en esta clase, implicando ajustar la base de imágenes para tener mayor información de entrenamiento y un análisis exhaustivo por parte de los expertos para llegar a determinar un cribado acertado de las imágenes con etiqueta equivocada. Por último, el área promedio indica un sistema robusto para tener una alta clasificación de las seis clases.

# Capítulo 5

## Conclusiones y Trabajos Futuros.

---

En este capítulo se presenta las conclusiones a las que se ha llegado sobre los principales temas que constituyen la metodología implementada en los capítulos anteriores. De igual forma se plantean los futuros trabajos de investigación con el fin de mejorar los resultados.

### 5.1 Conclusiones

Este trabajo presentó una evaluación de la metodología *Bag of Features* para la clasificación de células escamosas no queratinizada de muestras citológicas teniendo en cuenta el estándar de malignidad Bethesda 2001. El esquema propuesto está determinado por algoritmos en la detección y extracción de características locales, denominados operadores de puntos de interés. Seguido de la creación del *codebook* por parte del algoritmo *K-means++* se utilizó un método de representación de imágenes en forma de histograma usando palabras visuales o *keypoints*. Finalmente, se hace uso del algoritmo de clasificación SVM con un kernel RBF y una búsqueda automática de los hiper-parámetros con un ajuste de validación cruzada, encontrando el mejor modelo de clasificación.

Mediante tres experimentos se demostró la efectividad del enfoque. En la clasificación de las clases Normal y Anormal se alcanzó valores de las métricas de precisión, *recall* y F1, superiores al 95% con una combinación *SURF/OpponentSURF*, la razón principal es la descripción de la información de la intensidad de color, y en consecuencia se obtiene un modelo con alto índice de probabilidad de clasificar las imágenes correctamente, comprobado con un 95.68% de AUC. Este rendimiento es similar en la combinación *SIFT/OpponentSIFT*, debido a un alto grado de diferenciación de información de las intensidades de color, no obstante, la dimensión de los descriptores o vectores de características presentan un incremento considerable, causando mayor tiempo de procesamiento. Cabe resaltar que las mejores representaciones de las imágenes se alcanzaron con tamaños de *codebook* igual a 300, 500, y 900, para *SURF/OpponentSURF* y 1000 para *SIFT/OpponentSIFT* garantizando una clasificación acertada de la totalidad de las muestras.

Por otra parte, en el experimento dos y tres donde se utilizó la totalidad de las clases planteadas en el objetivo general. Se logró una precisión de 79.9%, 78.7% en *recall* y 78.36% de F1, para el experimento dos, mientras que para la prueba tres se alcanzó una precisión del 80.63%, 79.81% en *recall* y 79.45% en F1, con combinaciones de *SIFT/SIFT* y

*SIFT/OpponentSIFT* respectivamente, sin embargo, en el último experimento, la descripción de las imágenes no presenta un rendimiento de gran diferencia con respecto al resto de las métricas de las tres combinaciones, esto se debe a que los descriptores tipo *early-fusión* con características de color no caracterizaron adecuadamente las intensidades de tonalidad en las clases anormales contiguas. No obstante, se logró clasificar correctamente la mayoría de imágenes. El número de clústeres para los mejores resultados fue de 500 para la prueba dos, 400 y 1000 en el experimento con el conjunto balanceado de seis clases.

Se resalta el uso de los operadores de puntos de interés para la detección y descripción de información en imágenes con características similares, como las citológicas, y bajo el enfoque de representación *Bag of Features*. Con base a las características locales obtenidas con invariantes a escala y rotación, el modelo de clasificación en SVM con kernel RBF, es lo suficientemente discriminativo para lograr clasificar correctamente la mayoría de imágenes.

Una de las principales ventajas del enfoque BoF es su adaptabilidad a diversas colecciones de imágenes, no solo en escenas simples. Por el contrario, como se evidenció en los resultados del trabajo, el rendimiento del esquema es robusto bajo condiciones de información visual de contenido semejante que requiere de expertos para llegar a diferenciar. Finalmente, se concluye que el esquema propuesto presenta un marco flexible, ajustable para diversas técnicas, en cada una de las etapas, ya sea en la detección y descripción de las características locales, en la construcción del *codebook* con diferentes algoritmos de agrupamiento, igualmente para el procedimiento de clasificación.

## 5.2 Trabajos Futuros

Los trabajos futuros para el mejoramiento incluyen una evaluación más exhaustiva en algoritmos de detección y extracción de características de puntos de interés. Esta evaluación también incluirá una comparación de técnicas de clasificación, además, se debe tener en cuenta el diagnostico de varios expertos para obtener información más precisa del conjunto de imágenes.

## ANEXOS.

---

# Capítulo A

## Manual de Usuario.

---

El manual de usuario tiene como propósito, brindar de manera explícita y específica una guía de los procedimientos llevados a cabo para el desarrollo y funcionamiento óptimo de la aplicación desarrollada, teniendo en cuenta las diferentes librerías y herramientas utilizadas. Así mismo como servir de instrumento de apoyo y mejora de la aplicación. El manual se divide en dos secciones.

- En la primera sección se describe la correcta instalación y configuración de *OpenCV-Python* en el sistema operativo Windows 10 con sus respectivas librerías, además la instalación de *pyQt* herramienta utilizada para la creación de la interfaz de usuario.
- En la segunda sección se describe la aplicación, y el uso de la interfaz gráfica de usuario.

### A.1 Instalación y configuración de librerías

A continuación, se procede a descargar *Python* de la página oficial <https://www.python.org/downloads/>, se escogió la versión 2.7.13 por ser la versión más estable. Se debe revisar el tipo de arquitectura del sistema operativo del equipo (32 bits o 64 bits). Se ejecuta el archivo descargado y se presiona siguiente (ver Figura A. 1, Figura A. 2).

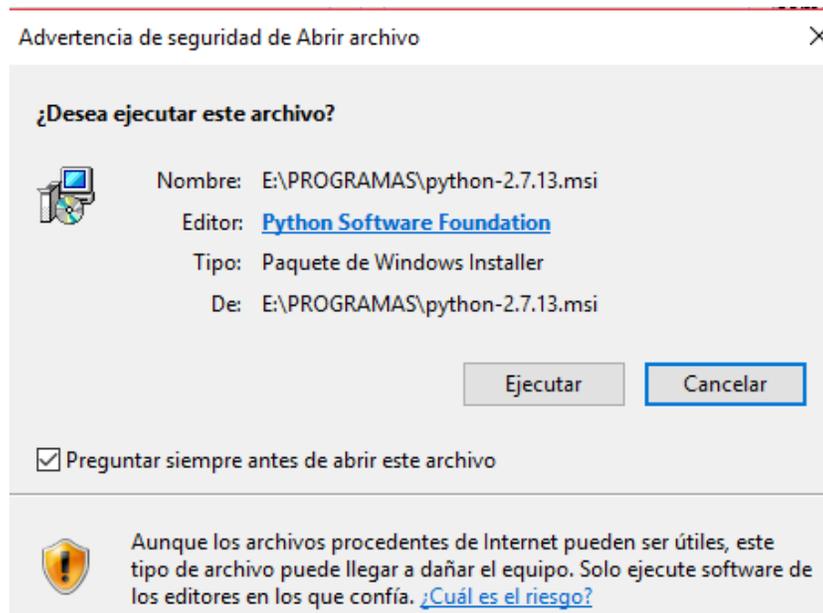


Figura A. 1 Ventana de ejecución del instalador Python.

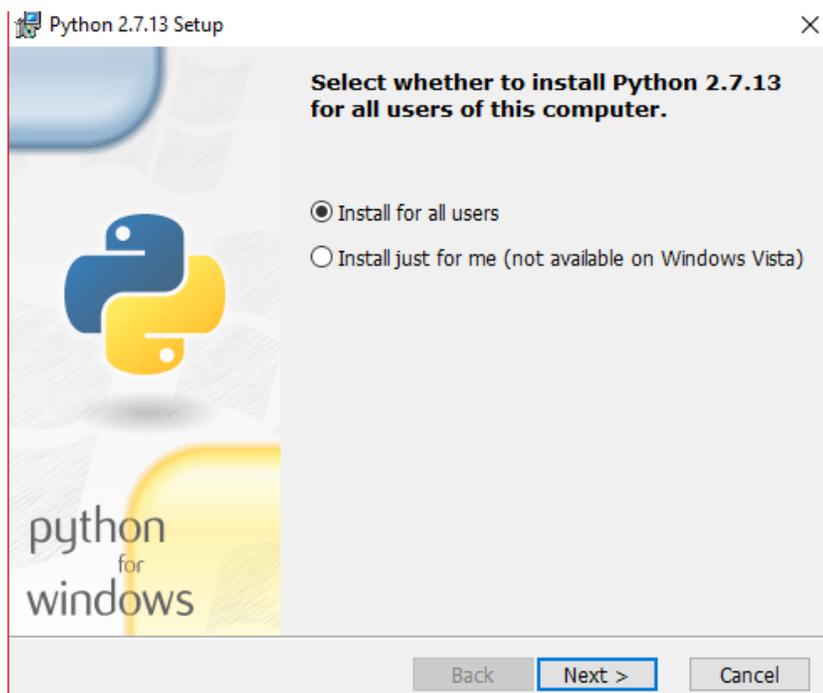
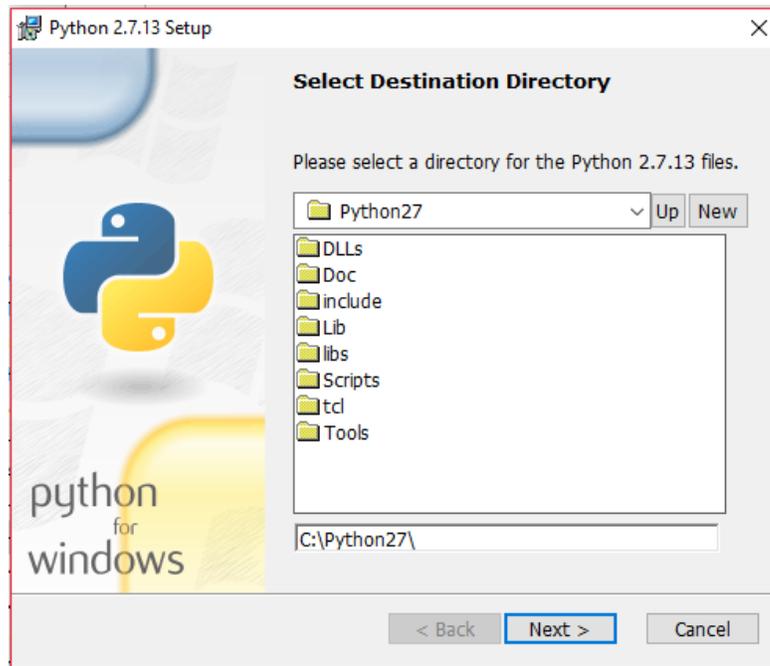


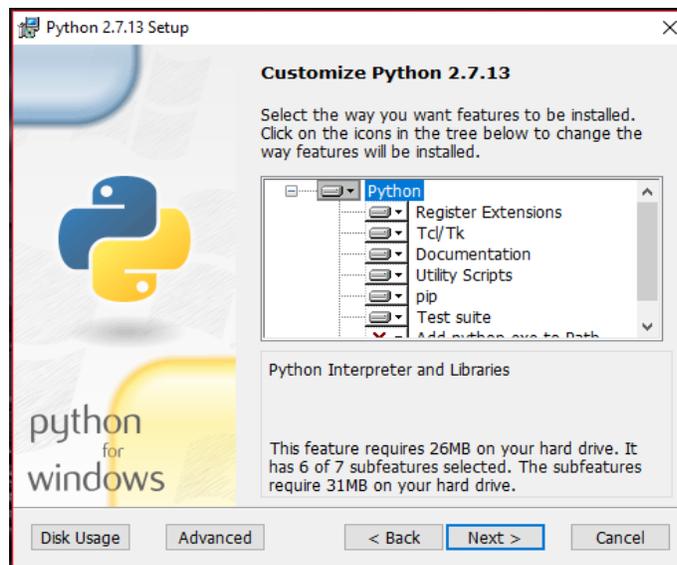
Figura A. 2: Selección de usuarios.

Se debe seleccionar la carpeta C:\ como ruta de destino (ver Figura A. 3)



*Figura A. 3: Selección de directorio de destino.*

En la ventana de personalización se deben instalar todos los complementos que aparecen en pantalla (*Figura A. 4*).



*Figura A. 4: ventana de personalización Python.*

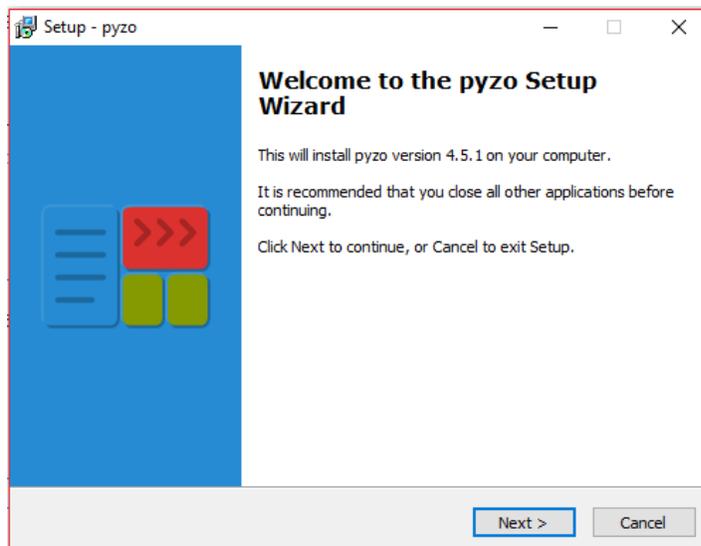
Una vez realizado esto, se presiona siguiente y se espera hasta que termine la instalación, para cerrar la ventana de instalación se presiona en finalizar (ver *Figura A. 5*).



*Figura A. 5: Ventana de finalización Python.*

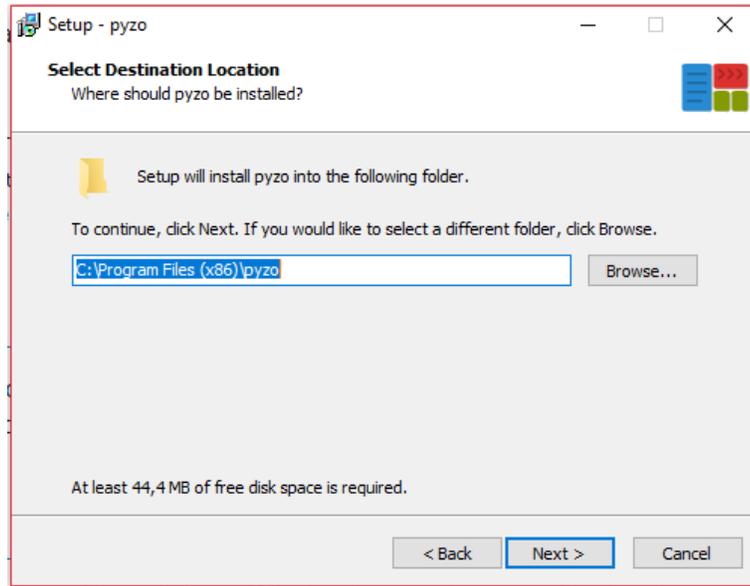
A pesar, de que **Python** dentro sus paquetes de instalación tiene su propio IDE (entorno de desarrollo interactivo), se utilizó para una mayor interacción y facilidad de implementación del código, un IDE especializado en el lenguaje *Python* llamado *Pyzo*, el cual se puede descargar de su página oficial <http://www.pyzo.org/>.

Una vez descargado el archivo de instalación, se procede a ejecutar (ver Figura A. 6).



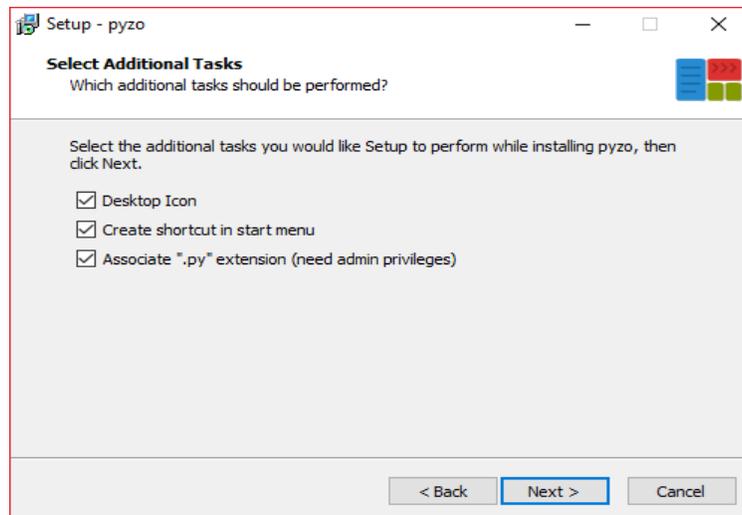
*Figura A. 6: Ventana de instalación de Pyzo.*

A continuación, se selecciona la ubicación de instalación y se presiona siguiente (ver Figura A. 7).



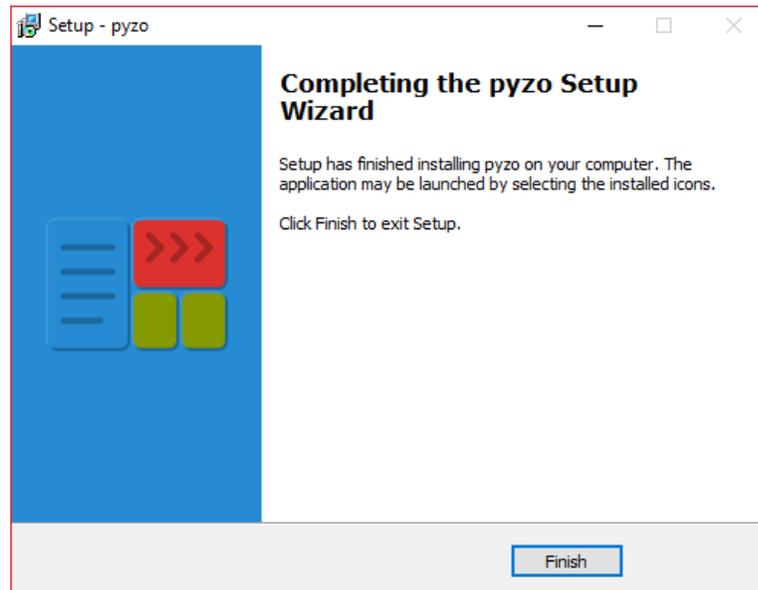
*Figura A. 7: Selección del destino de ubicación pyzo.*

En la selección de tareas adicionales se escoge a criterio propio lo que se considere pertinente configurar y se presiona siguiente (ver Figura A. 8).



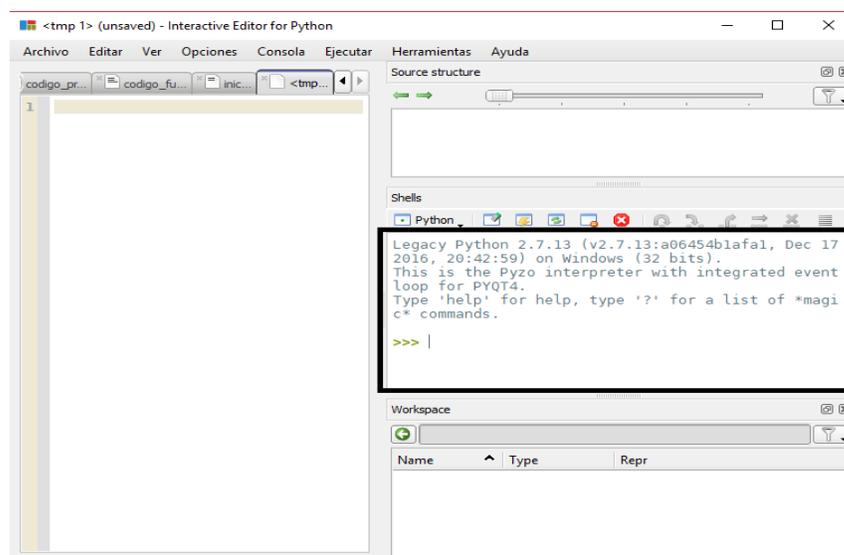
*Figura A. 8 Configuraciones adicionales pyzo.*

Una vez hecho esto, se espera a que termine la instalación y como último paso se presiona finalizar como se muestra en Figura A. 9.



*Figura A. 9: Ventana de finalización pyzo.*

Para la instalación de las librerías se abre el IDE instalado anteriormente y se sitúa el cursor en la pantalla subrayada tal y como aparece en la Figura A. 10.



*Figura A. 10: Ventana de inicio pyzo.*

A continuación, se digita el siguiente comando: *pip install + el nombre de la librería requerida* como se muestra en la siguiente imagen.

```
Kernel process terminated for restart. (0)

Legacy Python 2.7.13 (v2.7.13:a06454blafal, Dec 17 2016,
20:42:59) on Windows (32 bits).
This is the Pyzo interpreter with integrated event loop f
or PYQT4.
Type 'help' for help, type '?' for a list of *magic* comm
ands.

>>> pip install numpy
```

Figura A. 11: línea de comandos pyzo.

Para el buen y completo funcionamiento del trabajo realizado se requirieron las siguientes librerías:

- ✓ **NumPy** es una extensión de *Python*, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.
- ✓ **Scipy** es una biblioteca *open source* de herramientas y algoritmos matemáticos para *Python*, contiene módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, FFT, procesamiento de señales y de imagen, resolución de ODEs y otras tareas para la ciencia e ingeniería.
- ✓ **Matplotlib** es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o vectores en el lenguaje de programación *Python* y su extensión matemática *NumPy*.
- ✓ **Sklearn** (anteriormente *scikits.learn*) es una biblioteca de aprendizaje de máquina de software libre para el lenguaje de programación *Python*. Presenta varios algoritmos de clasificación, regresión y agrupación, incluyendo máquinas de vectores de soporte, bosques aleatorios, aumento de gradiente, k-medias y DBSCAN, y está diseñado para interoperar con las bibliotecas numéricas y científicas de *Python*, *NumPy* y *SciPy*.

Para comprobar la correcta instalación de las librerías, se digita en el *Shell* (línea de comandos) *import + el nombre de la librería*, el cual permite importar las librerías que aparecerán en el *workspace* sin ningún error tal y como aparece en la siguiente Figura A. 12.

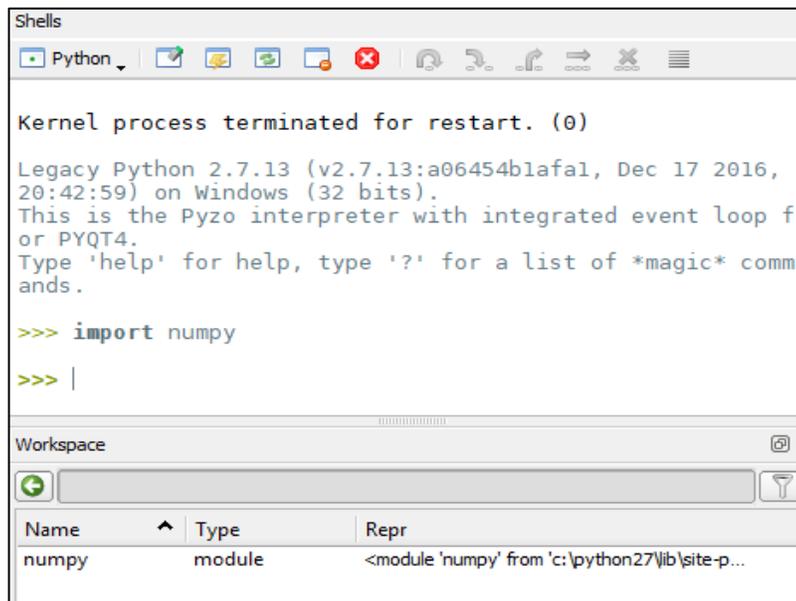


Figura A. 12: Importación de las librerías

Una vez se tienen todas las librerías de *Python* se procede a instalar *OpenCV*, que es la herramienta especializada y más utilizada para brindar solución a los problemas de visión de computadora. Para ello dirigirse a la página oficial <https://opencv.org/releases.html> y descargar la versión 2.4.13.6 que es la más reciente, actualizada y compatible para la versión de *Python* 2.7. Una vez descargada se abre el ejecutable, y se pedirá escoger la ruta de extracción (ver Figura A. 13).

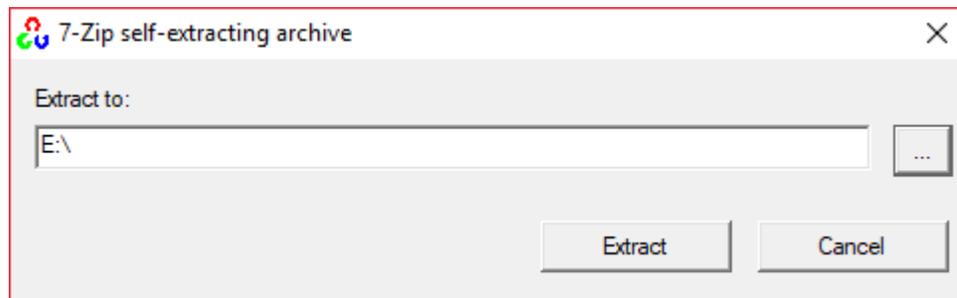


Figura A. 13: Ventana de escogencia de la ruta de extracción de *openCV*.

Al terminar la extracción, se ingresa a la siguiente ubicación `opencv/build/python/2.7` donde aparecerá dos carpetas con las arquitecturas disponibles del sistema operativo (32 o 64 bits), dentro de ellas se encuentra el archivo `cv2.pyd`, el cual se copiará y se pegará en la siguiente ruta `C:/Python27/lib/site-packages` como se muestra en la Figura A. 14.

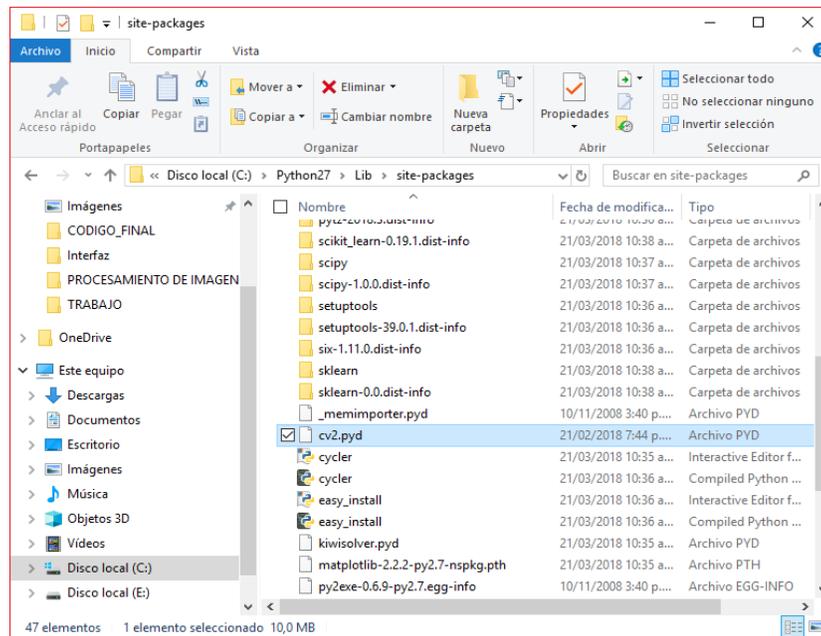


Figura A. 14: Ruta ubicación de la librería *openCV* en *Python*.

Por último, para verificar la correcta instalación de *OpenCV*, se abre el IDE y se digita el siguiente código en el terminal (ver Figura A. 15).

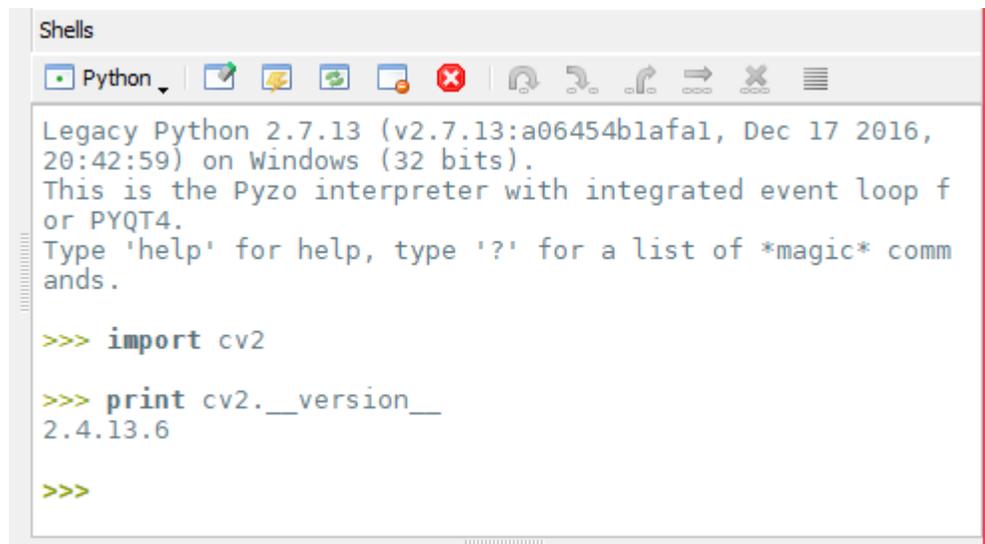
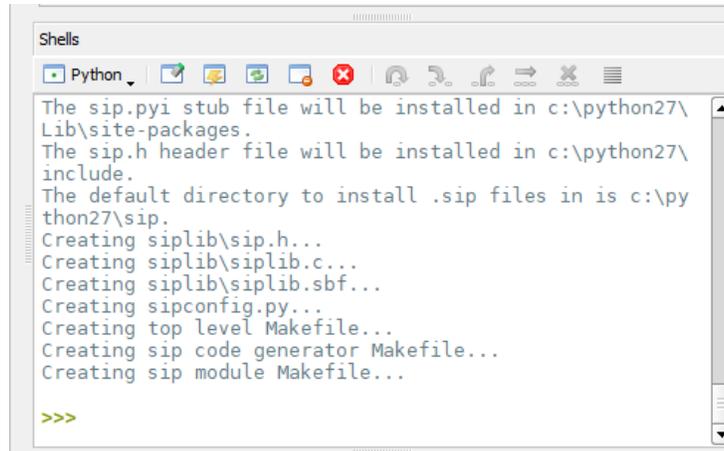


Figura A. 15: verificación de la librería *openCV* en el *Shell pyzo*.

Finalmente se lleva a cabo la instalación de *PyQT*. Antes se debe construir *SIP* descargándolo del siguiente enlace <https://www.riverbankcomputing.com/software/sip/download>. Una vez descargado se extrae en la ubicación que se desee y a continuación se abre el archivo

*configure.py* mediante el IDE *pyzo*. Para compilar el programa se debe presionar la combinación de teclas *Ctrl+Shift+E*, con el fin de que no aparezcan errores tal como se observa en la Figura A. 16.



```
Shells
Python
The sip.pyi stub file will be installed in c:\python27\
Lib\site-packages.
The sip.h header file will be installed in c:\python27\
include.
The default directory to install .sip files in is c:\py
thon27\sip.
Creating siplib\sip.h...
Creating siplib\siplib.c...
Creating siplib\siplib.sbf...
Creating sipconfig.py...
Creating top level Makefile...
Creating sip code generator Makefile...
Creating sip module Makefile...

>>>
```

Figura A. 16: Construcción del módulo SIP en *pyzo*.

1. Ya con SIP construido, se ingresa al siguiente enlace donde se encuentran los diferentes instaladores <https://sourceforge.net/projects/pyqt/files/PyQt4/PyQt-4.11.4/> se escoge el tipo de sistema operativo, la versión de *Python* instalada en este caso 2.7 y se procede a descargar.
2. Se ejecuta el archivo descargado y se presiona siguiente (ver Figura A. 17).

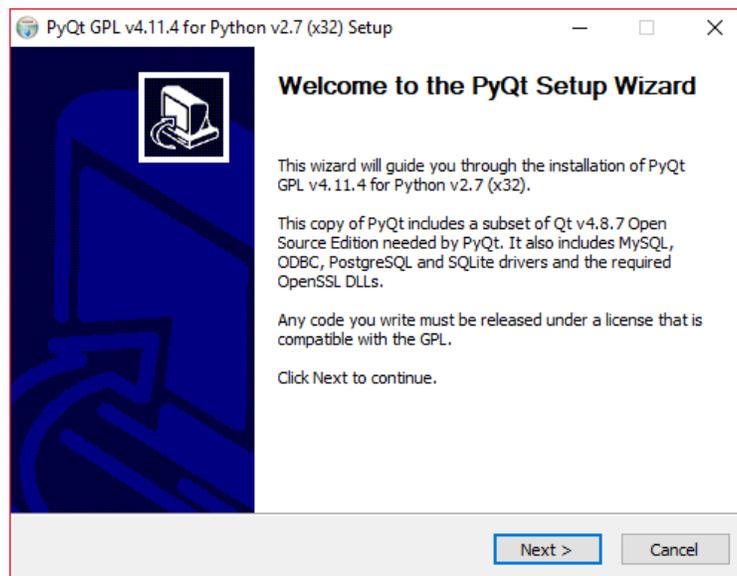


Figura A. 17: ventana de instalación *pyQT*.

3. Aceptar los acuerdos de licencia (ver Figura A. 18).

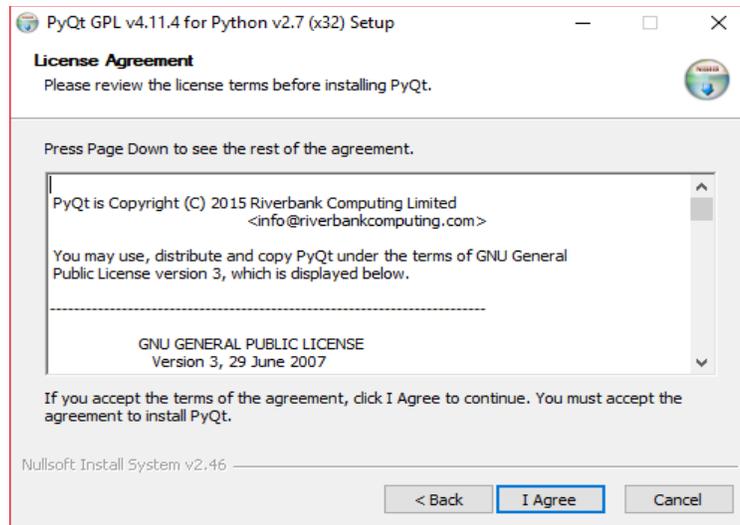


Figura A. 18: Acuerdos de licencia pyQT.

4. Se seleccionan todos los componentes y se presiona siguiente (ver Figura A. 19).

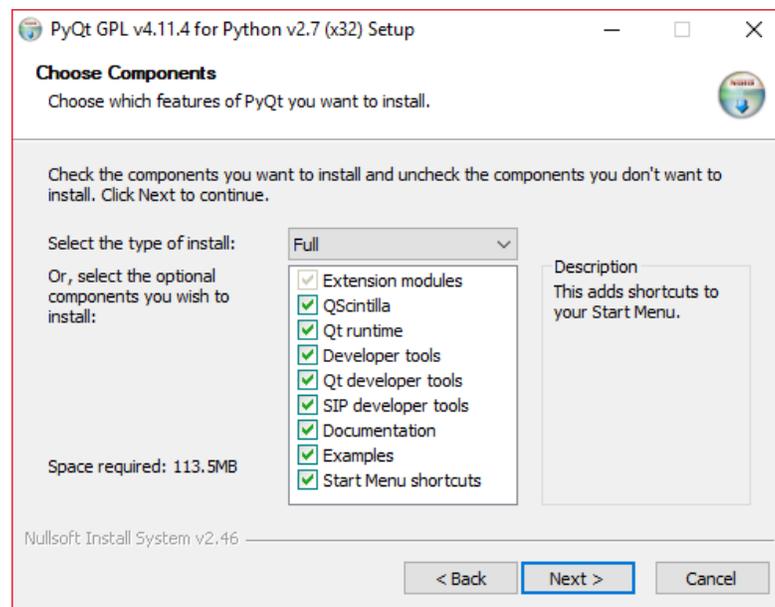
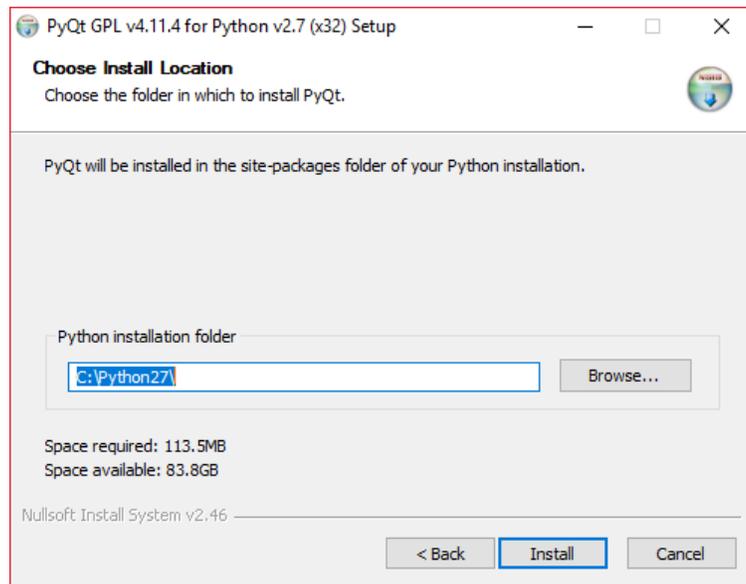


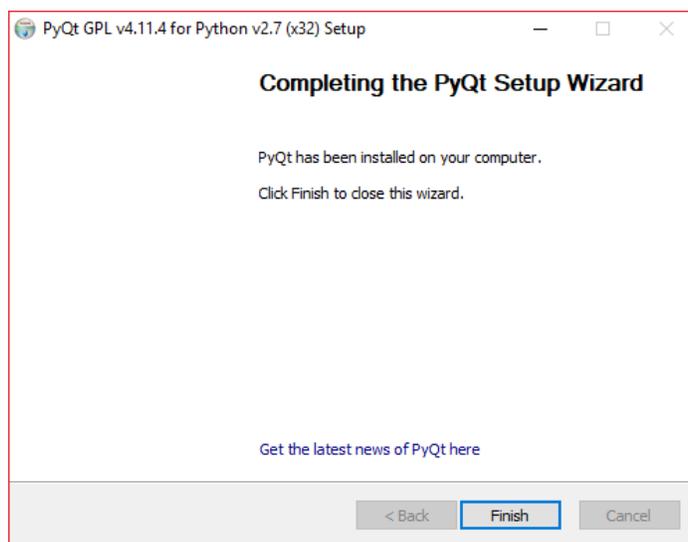
Figura A. 19: Escoger componentes pyQT.

5. Se escoge la misma ruta de ubicación donde se instaló Python (ver Figura A. 20).



*Figura A. 20: Ruta de instalación pyqt.*

7. Para terminar hacer clic en *Finish*.



*Figura A. 21: Ventana de finalización de instalación pyzo.*

## A.2 Manual de Uso

La aplicación desarrollada presenta una interfaz de fácil manejo, la cual está dividida en tres fases que son: la ventana de inicio, de entrenamiento y de evaluación; a continuación, se mostrarán las diferentes áreas de trabajo y las configuraciones de uso que permite la interfaz con un breve ejemplo.

- **Ventana de inicio:** es la ventana principal de la interfaz para su ejecución, debemos situarnos en la carpeta que contiene todos códigos del programa, a continuación, abrimos el código inicio.py con el *IDE Pyzo* como se muestra en la Figura A. 22.

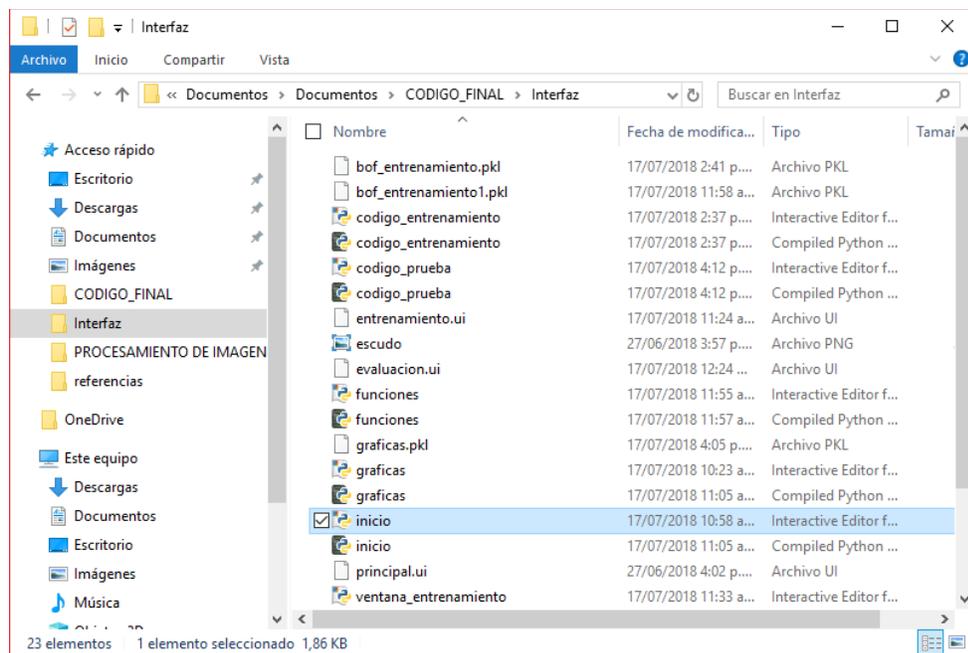


Figura A. 22 Carpeta con los diferentes códigos del trabajo.

Ya en el *IDE* se cuenta con dos opciones de ejecución, la primera opción se logra presionando la combinación de teclas *Ctrl+Shift+E*, y como segunda opción debemos situarnos en el panel de herramientas, seleccionar la opción ejecutar y a continuación presionar “Ejecutar el comando como un script” tal como se muestra en la Figura A. 23.

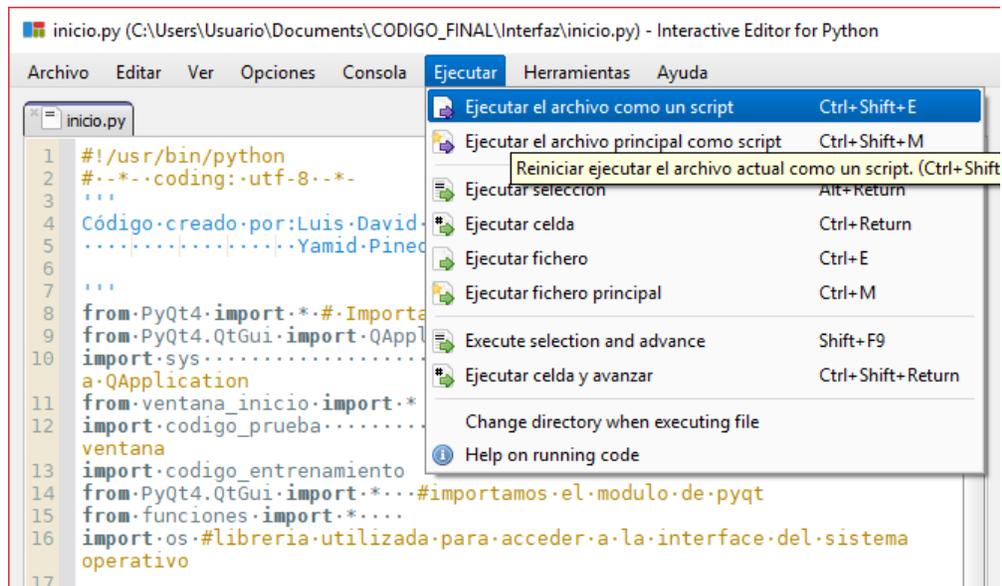


Figura A. 23 Ejecución del código inicio.py

Una vez lograda la ejecución de la ventana de inicio, se tendrán tres opciones a elegir, dos de ellas abrirán las ventanas complementarias de la aplicación que son la ventana de entrenamiento y evaluación respectivamente; como última opción se cuenta con el botón de salir que cerrara la aplicación (ver Figura A. 24).



Figura A. 24: Distribución ventana de inicio de la interfaz.

Es recomendable ejecutar el botón de entrenamiento para realizar la configuración deseada por el usuario, ya que si se ejecuta el botón de evaluación se cargará un modelo de entrenamiento de configuración predeterminado. En este caso se escogió como primera opción la etapa de entrenamiento que se describe a continuación.

• **Ventana de Entrenamiento:** contiene las posibles configuraciones para la ejecución de *bag of features*. Dentro de las configuraciones se encuentran la selección de los tipos de detector y descriptor, la cantidad de clústeres, las iteraciones de validación cruzada, la selección para la visualización de imágenes con *keypoints* y sin ellos y una tabla de resultados de los mejores parámetros de optimización.

Para la ejecución de la fase de entrenamiento es necesario seguir una serie de pasos como se ilustran en la Figura A. 25 y se describen a continuación:

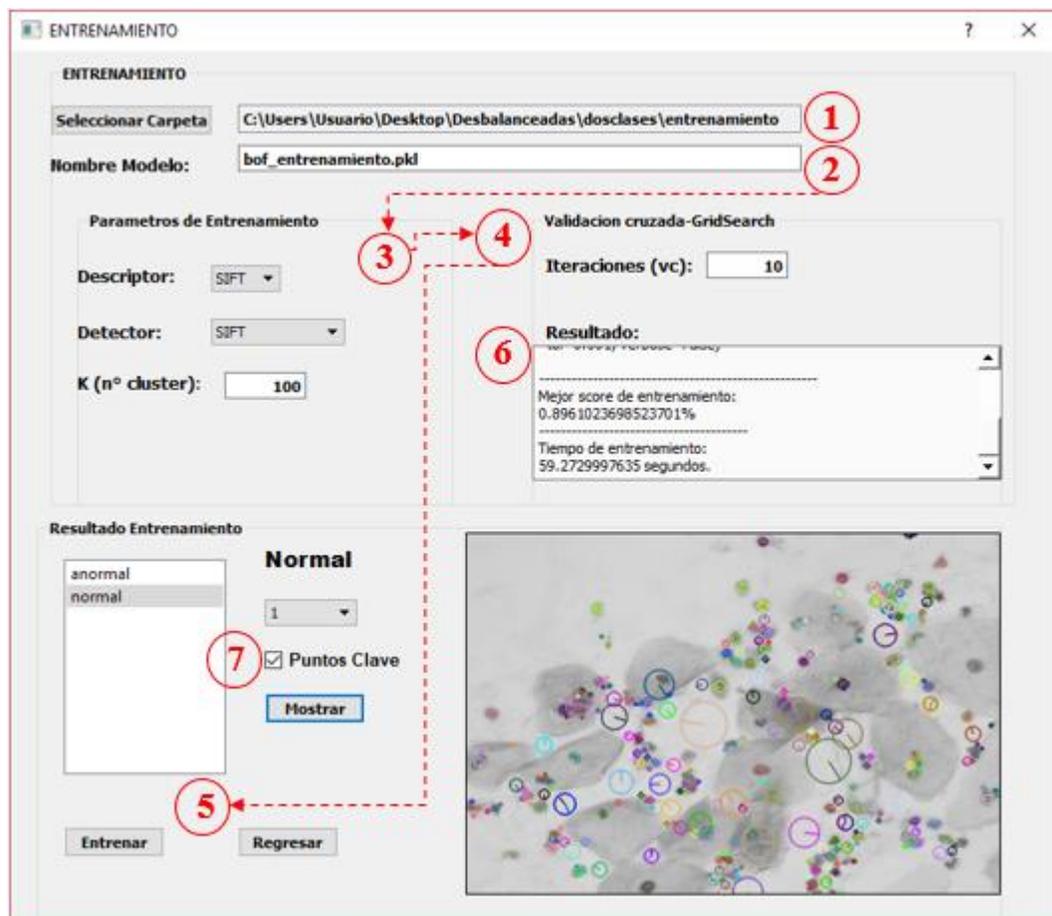


Figura A. 25: Distribución ventana de entrenamiento de la interfaz.

- 1) Se debe seleccionar la carpeta que contiene las imágenes citológicas con sus diferentes clasificaciones. Si se desea inspeccionar las imágenes cargadas en la interfaz, basta con seleccionar la clase, el número de la imagen y a continuación el botón mostrar.

- 2) Se da nombre al modelo, el cual es el archivo donde se almacenarán las diferentes variables del resultado de la etapa de entrenamiento, cabe resaltar que el nombre debe tener al final la extensión “. *pkl*”.
- 3) Para la etapa de extracción de características y agrupamiento, para ello se escoge el tipo de detector y descriptor deseado, además de que se debe digitar el número de clústeres deseado el cual por defecto aparecerá con el valor de 100.
- 4) Se debe digitar el número de veces que se ejecuta la validación cruzada para la selección de los hiper-parámetros.
- 5) Se debe presionar el botón entrenar, que ejecutará el código con las etapas anteriormente configuradas y habilitará la opción de visualización de las imágenes con sus respectivos *keypoints*.
- 6) En el cuadro de resultados, se visualizará los mejores parámetros seleccionados para el clasificador SVM, el mejor *score* de entrenamiento y el tiempo total de este.
- 7) Si se desea visualizar los *keypoints* encontrados, se debe seleccionar la opción “*puntos clave*” y el número de la imagen; una vez hecho esto se presionará el botón mostrar.
- 8) Como última opción procedemos a regresar al menú principal para continuar con la etapa de prueba y poder evaluar el rendimiento del trabajo realizado.

• **Ventana de Evaluación:** contiene las diferentes métricas del resultado de clasificación y además permite la visualización individual de las imágenes en las diferentes clases con la respectiva etiqueta de visualización. En la Figura A. 26 se ilustran las fases para la ejecución de la ventana de evaluación y se describen a continuación:

- 1) Al igual que la fase de entrenamiento, se debe seleccionar la carpeta que contenga las imágenes a ser evaluadas; una vez realizado se habilitará el botón de cargar modelo.
- 2) Se debe cargar el modelo que corresponda con el conjunto de imágenes anteriormente seleccionadas, es decir si el experimento pertenece a dos clases, a seis clases desbalanceadas o a seis clases balanceadas.
- 3) Se procede a ejecutar el botón de evaluar, el cual ejecutará la configuración y el algoritmo para la obtención de los resultados.
- 4) Una vez terminada la evaluación, en el cuadro de resultados se mostrará el tiempo que tardo en la realización y la exactitud del modelo de clasificación.

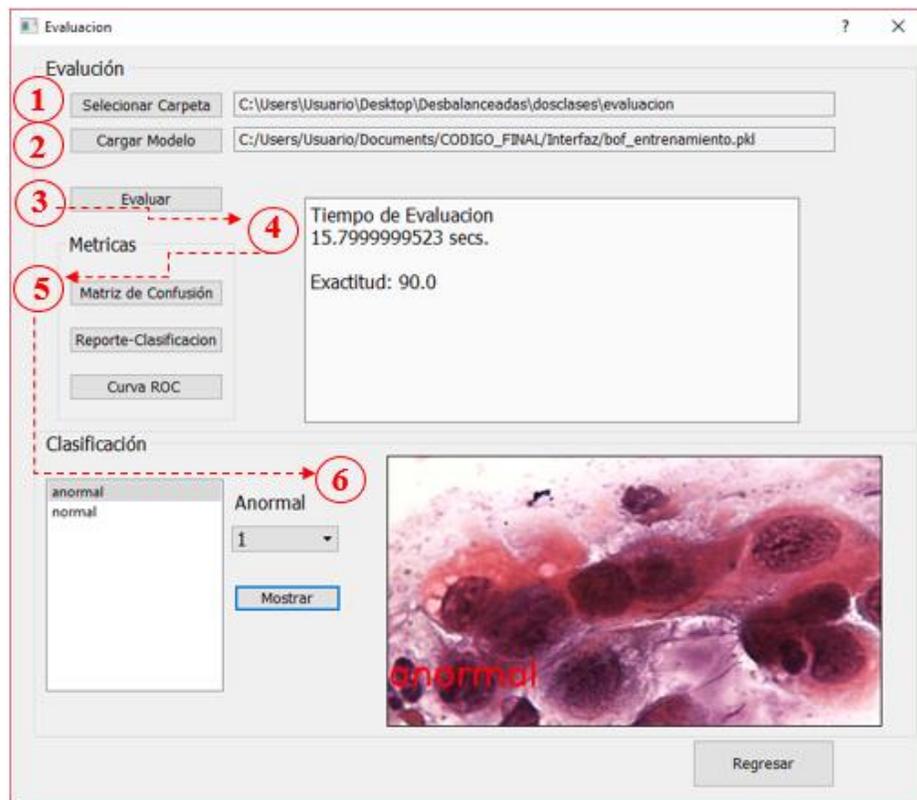


Figura A. 26: Distribución ventana de evaluación de la interfaz.

- 5) Se procede a la visualización de las demás métricas presentadas en el trabajo, como lo son: la gráfica de curva ROC AUC, la matriz de confusión y el reporte del clasificador.
- 6) Por último, también estará habilitado el botón “mostrar” que permitirá visualizar la respuesta cualitativa de cada una de las imágenes de evaluación con sus respectivas etiquetas.

### A.3 Manual de Implementación

A continuación, se expone la implementación de las diferentes funciones de Python usadas en las etapas de *Bag of Features*.

- **Lectura de Imágenes:** en este procedimiento se utilizó la función `glob.glob` (*dirección de archivo*) en la línea 57 y 59, el cual devuelve una lista de los nombres que se encuentran disponibles en la ruta; para la utilización de esta función se debe importar la librería `glob`, con la siguiente expresión: `import glob`. Con respecto a la ruta se debe especificar el tipo de extensión de los archivos, (+“/\* . jpg”) en este ejemplo se indica una extensión `jpg` de las imágenes.

- **Detección y Descripción de Características:** la extracción de características se obtuvo con las funciones SIFT y SURF, además de los descriptores OppenetSIFT y OppenentSURF. A continuación, se describen las funciones, pertenecientes a la librería **OpenCV 2.4.13.3**. Primero, se importa las librerías OpenCV y numpy: `import cv2, import numpy`, para el procesamiento de imágenes y de las operaciones matriciales, respectivamente. Seguido, se da lectura de una imagen o de un conjunto, a través, del comando `cv2.imread (conjunto de imágenes)`. El parámetro de salida corresponde a una matriz por imagen. Posterior, se pasa las imágenes a escalas de grises: `cv2.cvtColor(img,CV2.COLOR_BGR2GRAY)`, el primer parámetro de entrada corresponde al conjunto de imágenes leídas, y el parámetro `CV2.COLOR_BGR2GRAY` cambia el espacio de colores a escalas de grises. Finalmente, se crea el detector (`Detector`) y descriptor (`Descriptor`) para obtener los puntos clave (`kpts`) y describir las características (`des`).

```

: // Código Funciones
78 Detector = cv2.FeatureDetector_create(detector_type)
79 Descriptor=cv2.DescriptorExtractor_create(descriptor_type)
82 kpts=detector.detect(gray)
86 kpts,des=descriptor.compute(gray,kpts)

```

En el uso de descriptores OppenetSIFT y OppenentSURF se omite el cambio a escala de grises. En siguiente enlace se describe en detalle las funciones para la extracción de características.

[https://docs.opencv.org/2.4/modules/features2d/doc/common\\_interfaces\\_of\\_descriptor\\_extractors.html](https://docs.opencv.org/2.4/modules/features2d/doc/common_interfaces_of_descriptor_extractors.html).

- **Construcción del Codebook:** en esta etapa se utilizó el algoritmo de clustering K-means, por medio del conjunto de funciones KMeans, adscrito a la librería de *Scikit-Learn*, utilizado para el aprendizaje automático. Primero se realiza la configuración de los parámetros, `kmeans = KMeans(parámetros de configuración)` (ver el enlace:<http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>). Una vez obtenido la configuración, se realiza el cálculo del agrupamiento con el comando: `kmeans.fit(matriz de descriptores)`, donde el parámetro de entrada es la matriz de todos los descriptores .

- Representación BoF:** en esta etapa se construye el histograma contando el número de ocurrencias de cada palabra visual, sin embargo, primero se asigna una etiqueta de las muestras a cada clúster, realizando la comparación de las distancias de cada descriptor con todos los centroides o representantes del codebook, de esta forma el descriptor queda asignado con la etiqueta del centroide más cercano. Esto se logra con la función `vq.vq(obs, codebook)`, donde los parámetros de entrada corresponde al conjunto de descriptores de cada imagen (`obs`) y el codebook obtenido en la etapa anterior. Seguido se calcula la ponderación con la función `tf-idf`. Se configura la función con el comando: `tf-idf = TfidfTransformer(parámetros de configuración)`, finalmente se realiza el cálculo con la expresión: `tf-idf.fit_transform(out_hist)`, donde el parámetro de entrada es el histograma. Para la utilización de la anterior expresión se debe incluir al inicio del *scrip* la línea: `from sklearn.feature_extraction.text import TfidfTransformer`
- Clasificación:** el clasificador utilizado en este trabajo es SVM. Se comienza con una búsqueda exhaustiva de los parámetros del modelo por medio de la técnica *GridSearch* en conjunto con la *validación cruzada*. El comando `GridSearchCV(parámetros de ajuste)` toma como entradas relevantes el conjunto valores de los parámetros de ajuste de clasificación, el tipo de estimador (en este caso SVM), la configuración de la validación cruzada y el tipo de puntuación con que se va evaluar (*recall\_macro*, *accuracy* o *precisión\_macro*), sin embargo, existen más parámetros, detallados en el siguiente enlace [http://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.GridSearchCV.html](http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html). La configuración de la técnica de validación cruzada se logra con la función `StratifiedShuffleSplit(parámetros entrada)`, donde los parámetros de entrada está definida por la cantidad de repeticiones de reorganización y división, el porcentaje del conjunto de datos de entrenamiento y evaluación. lo anterior se resumen en las siguientes líneas de código.

```

: // Ejemplo de Entrenamiento SVM
1 C_range = np.logspace(-5,15, num=8,base=2)
2 lambda_range = np.logspace(-15,3, num=8,base=2)
3 folds=StratifiedShuffleSplit(n_splits = 10,test_size = 0.2
,random_state = 4),
4 parametros = [{'kernel': ['rbf'],'gamma':lamda_range , 'C':
C_range,'class_weight':['balanced']}]
5 par_opt = GridSearchCV(svm.SVC(), parametros, cv=folds, scoring =
'accuracy')

```

Una vez se configura la técnica *GridSearch* se procede a calcular los parámetros con el comando: `out_GridSearchCV.fit(parámetro de entrada)`(siguiendo el ejemplo: `best_par=par_opt.fit(parámetros de entrada)`). Los parámetros de entrada es la salida de la etapa de representación BoF y el vector de identificación de clases. Con el modelo obtenido se predice la clasificación del conjunto de imágenes de evaluación a través de la variable que contiene los mejores parámetros de clasificación y la expresión *predict*, es decir: `best_par.predict(parámetro de entrada)`. El parámetro de entrada corresponde a la representación ponderada con la función tf-idf de las imágenes de evaluación.

- **Métricas de Rendimiento:** las cinco métricas se obtienen con las siguientes líneas de código:

**Reporte de clasificación:** entrega las métricas de Precisión, Recall y F1-score para todas las clases. Para el uso de la función que permite el calculo de dichas métricas, antes se debe incluir la expresión: `from sklearn.metrics import classification_report`

```
: // Reporte de clasificación
1 from sklearn.metrics import classification_report
2 print (classification_report(labelsEval, predic, names=imEntre))
```

El primer parámetro de entrada (`labelsEval`) hace referencia al vector con el número de identificación de cada clase, seguido se tiene el parámetro `predic`, el cual corresponde a las predicciones realizadas por el modelo de clasificación. el ultimo parámetro de denominado `names` indica el vector con los nombres de cada clase.

**Matriz de confusión:** este tipo de métrica pertenece a la librería *scikit learn* por lo tanto se debe añadir la siguiente línea de código: `from sklearn.metrics import confusion_matrix`. Seguido se utiliza la función que permite calcular la matriz de confusión multiclase, la cual es: `confusión_matrix(y_true,y_pred)`, los parámetros de entrada más relevantes son los valores objetivos o verdaderos y el conjunto de valores estimados por el clasificador. En el siguiente enlace se encuentra más información:

[http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion\\_matrix.html#sklearn.metrics.confusion\\_matrix](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html#sklearn.metrics.confusion_matrix)

Por último, se usa la función `plot_confusion_matrix` (parámetros de entrada), para graficar los resultados de cada celda acompañado con una intensidad de color, de esta forma se resalta los mejores valores. El conjunto de líneas de código que permite esta tarea esta proporcionado en el enlace: [http://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_confusion\\_matrix.html](http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html).

**Curva ROC-AUC:** para esta última métrica se requiere importar las funciones de cálculo (`roc_curve` y `auc`) de la librería que las soporta (*scikit learn*), con la siguiente línea de código: `from sklearn.metrics import roc_curve, auc`. Cabe resaltar que la función `roc_curve` esta implementado para una clasificación binaria, por lo tanto, se debe realizar un proceso de binarización de las etiquetas en un tipo de enfoque de clasificación multiclase *one-vs-all*. Este proceso se muestra en el enlace: [http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.label\\_binarize.html#sklearn.preprocessing.label\\_binarize](http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.label_binarize.html#sklearn.preprocessing.label_binarize)

La función `roc_curve(y_true, y_score)`, tiene dos parámetros importantes de entrada. La primera (`y_true`) corresponde al vector de identificación binarizadas con valores verdaderos de las clases y `y_score` son los valores predichos binarizados. Finalmente, con los valores de las dos variables de salida, la tasa de verdaderos positivos (`tvp`) y tasa de falsos positivos (`tfp`), se calcula del índice AUC, con la siguiente expresión: `roc_auc[i]= auc (tvp[i], tfp[i])`.

# BIBLIOGRAFÍA

- [1] Z. Lu *et al.*, “Evaluation of Three Algorithms for the Segmentation of Overlapping Cervical Cells,” *IEEE J. Biomed. Heal. Informatics*, vol. 21, no. 2, pp. 441–450, 2017.
- [2] P. J. Schüffler, D. Schapiro, C. Giesen, H. A. O. Wang, B. Bodenmiller, and J. M. Buhmann, “Automatic single cell segmentation on highly multiplexed tissue images,” *Cytom. Part A*, vol. 87, no. 10, pp. 936–942, 2015.
- [3] A. Gençtav, S. Aksoy, and S. Önder, “Unsupervised segmentation and classification of cervical cell images,” *Pattern Recognit.*, vol. 45, no. 12, pp. 4151–4168, 2012.
- [4] “SEGMENTATION OF OVERLAPPING CERVICAL CELLS : A VARIATIONAL METHOD WITH STAR-SHAPE PRIOR Masoud S . Nosrati and Ghassan Hamarneh Medical Image Analysis Lab , School of Computing Science.”
- [5] L. Zhao, K. Li, J. Yin, Q. Liu, and S. Wang, “Complete three-phase detection framework for identifying abnormal cervical cells,” *IET Image Process.*, vol. 11, no. 4, pp. 258–265, 2017.
- [6] D. Kashyap, A. Somani, and J. Shekhar, “Cervical Cancer Detection And Classification Using Independent Level Sets And Multi SVMs,” *39th Int. Conf. Telecommun. Signal Process.*, pp. 523–528, 2016.
- [7] A. Bhan, G. Vyas, and S. Mishra, “Supervised segmentation of overlapping cervical pap smear images,” *2016 Int. Conf. Signal Process. Commun.*, vol. c, pp. 225–228, 2016.
- [8] A. Automática and D. I. Médicas, “Universidad Nacional de Colombia Representación de Bolsa de Características,” 2011.
- [9] J. C. Caicedo, A. Cruz, and F. A. Gonzalez, “Histopathology Image Classification Using Bag of Features and Kernel Functions BT - Artificial Intelligence in Medicine: 12th Conference on Artificial Intelligence in Medicine, AIME 2009, Verona, Italy, July 18-22, 2009. Proceedings,” pp. 126–135, 2009.
- [10] E. Nowak, “Sampling Strategies for Bag-of-Features,” pp. 490–503, 2006.
- [11] S. Keypoints and D. G. Lowe, “Distinctive Image Features from,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
- [12] K. Bora, M. Chowdhury, L. B. Mahanta, M. K. Kundu, and A. K. Das, “Automated classification of Pap smear images to detect cervical dysplasia,” *Comput. Methods Programs Biomed.*, vol. 138, pp. 31–47, 2017.
- [13] P. Velvizhy, Pavithra, and A. Kannan, “Automatic food recognition system for diabetic patients,” *6th Int. Conf. Adv. Comput. ICoAC 2014*, pp. 329–334, 2015.

- [14] J. C. Caicedo and E. Izquierdo, “Combining Low-level Features for Improved Classification and Retrieval of Histology Images,” *Trans. Mass-Data Anal. Images Signals*, vol. 2, pp. 68–82, 2010.
- [15] S. O’Hara and B. A. Draper, “Introduction to the Bag of Features Paradigm for Image Classification and Retrieval,” no. July, pp. 1–25, 2011.
- [16] S. H. Raza *et al.*, “Automated classification of renal cell carcinoma subtypes using bag-of-features,” *2010 Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBC’10*, pp. 6749–6752, 2010.
- [17] J. Scharcanski and M. E. Celebi, *Computer Vision Techniques for the Diagnosis of Skin Cancer*. 2014.
- [18] J. Barker, A. Hoogi, A. Depeursinge, and D. L. Rubin, “Automated classification of brain tumor type in whole-slide digital pathology images using local representative tiles,” *Med. Image Anal.*, vol. 30, pp. 60–71, 2016.
- [19] F. Jurie and B. Triggs, “Creating Efficient Codebook for Visual Recognition,” *Proc. {IEEE} Int. Conf. Comput. Vis.*, pp. 604–610, 2005.
- [20] D. Nistér and H. Stewénus, “Scalable recognition with a vocabulary tree,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, pp. 2161–2168, 2006.
- [21] T. Tommasi, F. Orabona, and B. Caputo, “CLEF2007 image annotation task: An SVM-based cue integration approach,” *CEUR Workshop Proc.*, vol. 1173, 2007.
- [22] K. Teeyapan, N. Theera-Umpon, and S. Auephanwiriyakul, “Application of support vector based methods for cervical cancer cell classification,” *2015 IEEE Int. Conf. Control Syst. Comput. Eng.*, no. November, pp. 514–519, 2015.
- [23] J. Uddin, F. N. Arko, N. Tabassum, T. R. Trisha, and F. Ahmed, “Bangla Sign Language Interpretation using Bag of Features and Support Vector Machine,” no. December, pp. 7–9, 2017.
- [24] A. Wiliem, Y. Wong, C. Sanderson, P. Hobson, S. Chen, and B. C. Lovell, “Classification of Human Epithelial type 2 cell indirect immunofluorescence images via codebook based descriptors,” *Proc. IEEE Work. Appl. Comput. Vis.*, pp. 95–102, 2013.
- [25] H. O. Ilhan and E. Celik, “The mesothelioma disease diagnosis with artificial intelligence methods,” *2016 IEEE 10th Int. Conf. Appl. Inf. Commun. Technol.*, pp. 1–5, 2016.
- [26] C. L. Pelea and J. F. González, “Citología Ginecológica: de Papanicolaou a Bethesda,” p. 206, 2003.
- [27] “Cáncer de cuello uterino (PDQ®)—Versión para pacientes - National Cancer Institute.” [Online]. Available: <https://www.cancer.gov/espanol/tipos/cuello-uterino/paciente/tratamiento-cuello-uterino-pdq>. [Accessed: 07-Jul-2018].

- [28] “<http://MedicoModerno.Blogspot.com>.”
- [29] B. Ghanbarzadeh and H. Almasi, “World ’ s largest Science , Technology & Medicine Open Access book publisher c,” *RFID Technol. Secur. Vulnerabilities, Countermeas.*, pp. 75–100, 2013.
- [30] A. Ng, “1. Supervised learning,” *Mach. Learn.*, pp. 1–30, 2012.
- [31] S. Mustafa, A. B. Dauda, and M. Dauda, “Image Processing and SVM Classification for Melanoma Detection,” 2017.
- [32] B. Kieffer, M. Babaie, S. Kalra, and H. R. Tizhoosh, “Convolutional Neural Networks for Histopathology Image Classification: Training vs. Using Pre-Trained Networks,” 2017.
- [33] S. Goswami and L. K. P. Bhaiya, “Brain Tumour Detection Using Unsupervised Learning Based Neural Network,” *2013 Int. Conf. Commun. Syst. Netw. Technol.*, pp. 573–577, 2013.
- [34] K. Roy, G. Subhramanya, and V. R. K. Rao, “ART based clustering of bag-of-features for image classification,” *2012 5th Int. Congr. Image Signal Process. CISP 2012*, no. Cisp, pp. 841–846, 2012.
- [35] F. F. Data, “Image Reconstruction from,” vol. 72, no. 6, 1984.
- [36] P. Ghosh, S. Antani, L. R. Long, and G. R. Thoma, “Review of medical image retrieval systems and future directions,” *2011 24th Int. Symp. Comput. Med. Syst.*, no. June 2014, pp. 1–6, 2011.
- [37] N. C. Mhala and S. H. Bhandari, “Improved approach towards classification of histopathology images using bag-of-features,” *2016 Int. Conf. Signal Inf. Process. IConSIP 2016*, pp. 2–6, 2017.
- [38] C. S. Venegas-Barrera and J. Manjarrez, “Visual Categorization with Bags of Keypoints,” *Rev. Mex. Biodivers.*, vol. 82, no. 1, pp. 179–191, 2011.
- [39] A. Cruz-Roa, J. C. Caicedo, and F. A. González, “Visual pattern analysis in histopathology images using bag of features,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5856 LNCS, pp. 521–528, 2009.
- [40] A. Cruz-Roa, J. C. Caicedo, and F. A. González, “Visual pattern mining in histology image collections using bag of features,” *Artif. Intell. Med.*, vol. 52, no. 2, pp. 91–106, 2011.
- [41] M. M. Duarte Villaseñor and L. Chang Fernandez, “Clasificación de objetos en imágenes usando SIFT,” 2010.
- [42] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes*

- Bioinformatics*), vol. 3951 LNCS, pp. 404–417, 2006.
- [43] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2564–2571, 2011.
  - [44] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 105–119, 2010.
  - [45] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “BRIEF: Binary Robust Independent Elementary Features,” pp. 778–792, 2010.
  - [46] A. Alahi, R. Ortiz, and P. Vandergheynst, “FREAK: Fast Retina Keypoint,” *2012 Ieee Conf. Comput. Vis. Pattern Recognit.*, pp. 510–517, 2012.
  - [47] B. Brisk and O. R. B. Card, “CV Reading BRISK : Binary Robust Invariant Scalable Keypoints,” pp. 1–8, 2011.
  - [48] M. Hussain, H. U. Rehman, O. Nazir, A. Hassan, and M. A. Dildar, “Separate Modal Analysis using Scale Invariant Feature Transform ( SIFT ) with Digital Image Elasto Tomography ( DIET ) for breast cancer screening test,” pp. 126–129.
  - [49] Q. Zhu, Y. Zhong, B. Zhao, G. Xia, and L. Zhang, “Bag-of-Visual-Words Scene Classifier With Local and Global Features for High Spatial Resolution Remote Sensing Imagery,” vol. 13, no. 6, pp. 747–751, 2016.
  - [50] R. Azhar, D. Tuwohingide, D. Kamudi, Sarimuddin, and N. Suciati, “Batik Image Classification Using SIFT Feature Extraction, Bag of Features and Support Vector Machine,” *Procedia Comput. Sci.*, vol. 72, pp. 24–30, 2015.
  - [51] A. Witkin, “Scale-space filtering: A new approach to multi-scale description,” *ICASSP '84. IEEE Int. Conf. Acoust. Speech, Signal Process.*, vol. 9, pp. 150–153, 1984.
  - [52] T. Lindeberg, “Feature Detection with Automatic Scale Selection,” *Int. J. Comput. Vis.*, vol. 30, no. 2, pp. 79–116, 1998.
  - [53] J. J. Koenderink, “The structure of images,” *Biol. Cybern.*, vol. 50, no. 5, pp. 363–370, 1984.
  - [54] L. H. Rios Gonzalez, “Sistema de Clasificación y Reconocimiento de Imágenes,” vol. 1, 2015.
  - [55] M. Brown and D. Lowe, “Invariant Features from Interest Point Groups,” *Br. Mach. Vis. Conf. BMVC 2002*, pp. 656–665, 2002.
  - [56] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001*, vol. 1, p. I-511-I-518, 2001.
  - [57] F. S. Khan, J. Van De Weijer, and M. Vanrell, “Top-down color attention for object

- recognition,” *Proc. IEEE Int. Conf. Comput. Vis.*, no. Iccv, pp. 979–986, 2009.
- [58] P. Ma, M. Seeland, M. Rzanny, N. Alaqraa, and J. Wa, “Plant species classification using flower images — A comparative study of local feature representations,” no. February, pp. 1–30, 2017.
- [59] S. Member, “Color Independent Components Based SIFT Descriptors for Object / Scene Classification,” no. 9, pp. 2577–2586, 2010.
- [60] K. Van De Sande, T. Gevers, and C. Snoek, “Evaluating color descriptors for object and scene recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1582–1596, 2010.
- [61] S. R. Dubey, P. Dixit, N. Singh, and J. P. Gupta, “Infected Fruit Part Detection using K-Means Clustering Segmentation Technique,” *Int. J. Interact. Multimed. Artif. Intell.*, vol. 2, no. 2, p. 65, 2013.
- [62] Y. Zhang, L. Gong, and Y. Wang, “An improved TF-IDF approach for text classification,” *J. Zhejiang Univ. Sci.*, vol. 6, no. 1, pp. 49–55, 2005.
- [63] R. Banaeeyan, M. H. Lye, M. F. A. Fauzi, H. A. Karim, and J. See, “Unsupervised face image retrieval using adjacent weighted component-based patches,” *Int. Conf. Intell. Adv. Syst. ICIAS 2016*, 2017.
- [64] O. Chapelle, “Choosing Multiple Parameters for Support Vector Machines,” *Mach. Learn.*, no. 46, pp. 131–159, 2002.
- [65] E. J. Carmona Suárez, “Máquinas de Vectores Soporte (SVM),” *Dpto. Intel. Artif. ETS Ing. Inforática, Univ. Nac. Educ. a Distancia*, pp. 1–25, 2014.
- [66] “Simple Tutorial on SVM and Parameter Tuning in Python and R | Machine Learning | HackerEarth Blog.” [Online]. Available: <https://www.hackerearth.com/blog/machine-learning/simple-tutorial-svm-parameter-tuning-python-r/>. [Accessed: 07-Jul-2018].
- [67] I. H. Mendez, “Análisis De Datos,” pp. 1–32.
- [68] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*, vol. 64, no. 9–12. 2007.
- [69] V. Bol, “Análisis de estrategias de clasificación multiclase en microarrays : relación con medidas de complejidad.”
- [70] J. I. Fernandez-Michelli, J. A. Areta, M. Hurtado, and C. H. Muravchik, “Polarimetric SAR image classification using EM method and G0p Model,” *2015 XVI Work. Inf. Process. Control*, pp. 1–6, 2015.
- [71] D. Mauricio, F. Carvajal, G. Andrés, and G. Gaitán, “Detección de grupos de fajillas en imágenes de paquetes de billete en diversas condiciones de iluminación y fondo mediante un clasificador SVM Strips groups detection in images of banknote packages

- in different lighting and background conditions using an ,” vol. 14, pp. 145–154, 2017.
- [72] U. Cauca, “Segmentación de imágenes mediante un algoritmo de agrupamiento espectral,” 2017.
- [73] J. H. Donis, “Evaluación de la validez y confiabilidad de una prueba diagnóstica (Assessment of the validity and reliability of a diagnostic test),” *Av. en Biomed.*, vol. 1, no. 2, pp. 73–81, 2012.
- [74] I. López and Pita, “Curvas ROC,” *Cad Aten Primaria*, vol. 5, no. 4, pp. 229–235, 1998.
- [75] A. Rocío, “Curvas ROC (Receiver-Operating-Characteristic) y sus aplicaciones,” pp. 12–16, 2017.
- [76] A. Torres Ortiz, “Curvas ROC para Datos de Supervivencia. Aplicación a Datos Biomédicos,” p. 79, 2010.
- [77] F. Herrera, “Edición de Conjuntos de Entrenamiento no Balanceados, haciendo uso de Operadores Genéticos y la Teoría de los Conjuntos Aproximados,” pp. 277–284.